MDA
ARCHITECTURE
MANAGEMENT
HUB

THE NATIONAL MARITIME DOMAIN AWARENESS ARCHITECTURE PLAN

A NATIONAL MARITIME INFORMATION SHARING ENVIRONMENT (MISE) IMPLEMENTED THROUGH COMMON DATA STANDARDS AND ARCHITECTURAL UNDERSTANDING.

Version 2.0, Release 2

THIS PAGE INTENTIONALLY LEFT BLANK

LETTER OF PROMULGATION

The *National Concept of Operations for Maritime Domain Awareness (MDA CONOPS)* directs the development of interagency and agency-specific policies, processes, procedures, and organizational relationships to align activities contributing to improving MDA. The MDA CONOPS identifies the Department of Defense/Department of Navy (DoD/DoN) as the agency responsible to define and document the processes required to achieve such an environment. The resulting Maritime Information Sharing Environment (MISE) is captured in this document.

Development of the net-centric principles outlined in this document was guided by the *MDA CONOPS*, the *Intelligence Reform and Terrorism Prevention Act of 2004*, and the *National Strategy for Information Sharing and Safeguarding*. Successful implementation will result in a secure, collaborative, information-sharing environment with unprecedented access to decision-quality information.

The services and capabilities contained in the attached document are immediately available. Goals and objectives achieved from implementation of the services and capabilities are applicable to all members of the Global Maritime Community of Interest.

RADM Samuel Cox

Attachment:

The National Maritime Domain Awareness Architecture Plan

This page intentionally left blank

ABOUT THIS DOCUMENT

The National Maritime Domain Awareness Architecture Plan describes a process for sharing maritime information. To provide the appropriate level of detail and technical specificity to the full spectrum of readers, the document is divided into multiple sections. They include:

Executive Summary

The *Executive Summary* is for senior leaders who want an overview of the plan. It summarizes major concepts and business processes.

Concept

The *Concept* is for mid-level managers and senior technical managers who need to understand, with some detail, the objectives, features and business processes of the plan.

Framework

The *Framework* is for technical managers or technology implementers who need to understand, with greater detail, the technical implementation and specifications of the plan.

Appendices

The *Appendices* are a collection of architectural and technical documents which are relevant to, but not otherwise included in, the National Maritime Domain Awareness Architecture Plan. Each appendix represents the state of the technology, process or definition at the time this plan was published.

Attachments

The *Attachments* are a collection of documents that describe the National Information Exchange Model – Maritime (NIEM-M) at the time this plan was published. While NIEM-M is not part of the plan, the definition of common data standards is integral to the success and implementation of this plan.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

A NATIONAL MARITIME INFORMATION SHARING ENVIRONMENT (MISE)
IMPLEMENTED THROUGH COMMON DATA STANDARDS AND
ARCHITECTURAL UNDERSTANDING.

> *"It is a national priority to efficiently, effectively, and appropriately share and safeguard information so any authorized individual (Federal, state, local, tribal, territorial, private sector or foreign partner) can prevent harm to the American people and protect national security. The Strategy points toward a future in which information support national security decision-making by providing the right information, at any time, to any authorized user, restricted only by law or policy, not technology; and where safeguarding measures, to include a comprehensive regimen of accountability, prevent the misuse of the information."*[1]

This passage from the *National Strategy for Information Sharing and Safeguarding* (the National Strategy) establishes the National vision for information sharing. The National Strategy goes on to identify three core information sharing principles:

1. Information as a National Asset
2. Information Sharing and Safeguarding Requires Shared Risk Management
3. Information Informs Decision Making

Guided by these principles and as directed by the National Concept of Operations for Maritime Domain Awareness (MDA CONOPS), the Department of Defense/Department of the Navy (DoD/DoN) led the effort to describe "an MDA architecture founded upon net-centric principles to provide a secure, collaborative, information-sharing environment"[2]. The resulting Maritime Information Sharing Environment (MISE) is captured in this plan. The MISE provides an internet accessible, unclassified information sharing capability where data providers and consumers manage and share maritime information through common data definitions and security attributes.

Being mindful of the principles identified in the National Strategy, the objectives in the MDA CONOPS and with full understanding that each participating agency has its own operational constraints; the MISE defines an operational and technical framework that enables information sharing by leveraging existing programs and systems. Within the U.S., agencies and organizations with maritime interests have their own requirements, authorities, information infrastructures, and resources. The MISE accounts for each of those constraints and defines a service oriented architectural approach that allows participation while protecting individual information and resources. Data and analytical products are shared via common data standards

[1] *National Strategy for Information Sharing and Safeguarding*, December 2012, Vision Statement
[2] *National Concept of Operations for Maritime Domain Awareness*, December 2007, Executive Summary

with access controls that allow data providers to manage their information within the constraints of their respective authorities or regulations.

The MISE employs industry standard service protocols that, in most cases, are already supported by participating agencies. The MISE *does not* define how agencies execute their missions, manage their infrastructure, or the products they develop. The MISE *does* define how the results of those mission activities or information products are made available to others. Through the definition of data standards within the National Information Exchange Model-Maritime (NIEM-M), the MISE provides a common vocabulary in four initial focus areas: Vessel Positions, Advance Notice of Arrival, Indicators and Notifications, and Maritime Operational Threat Response. While only four focus areas are defined in the initial effort, the standards and processes defined by the MISE are designed to be reusable and extensible to support future information sharing products and partners.

Currently, the concepts, capabilities, and standards outlined in this plan are being implemented across the U.S. Government. The Office of Naval Intelligence has adopted NIEM-M as the standard for future data services. The Navy Data Engineering Services Center (DESC) is a DoD organization supporting NIEM-M development. The technologies described in this plan have also been tested and implemented within multiple programs. The Coast Guard Interagency Operations Center project, the Air Force Maritime Domain Awareness System at MacDill AFB in Tampa, Florida and U.S. SOUTHCOM's Caribbean Sensor Information Integration are three programs operating today.

For the MISE to fulfill the principles outlined in the National Strategy, additional steps are required. The continued development of standards and the implementation of services from authoritative data sources are important first steps. Critical to the success is developing confidence in the processes and trust in the partnerships. The DESC has implemented the MISE in a commercial cloud environment for use by international, interagency or industry partners to develop and test the web services required to build that confidence and trust.

Information sharing is a mandated activity that directly affects National Security. However, information sharing efforts continue to be impeded by technical, legal, and policy concerns. The MISE defines a low cost, implementable solution while providing mechanisms to mitigate associated legal and policy concerns. While the MISE may not support the information sharing requirements of every agency, it will support the processing and management of the majority of the unclassified maritime information the nation collects, processes, and shares to support and improve our National Security in the Maritime Domain.

TABLE OF CONTENTS

CONCEPT

1. Background

The *National Strategy for Information Sharing and Safeguarding* (the National Strategy)and the *National Concept of Operations for Maritime Domain Awareness* (MDA CONOPS) describe an information sharing architecture founded upon data-centric principles to provide a secure, collaborative, information-sharing environment (ISE). The described ISE recognizes information as a national asset and safeguarding information is a shared responsibility.

The MDA CONOPS directed the development of three documents to define and describe the ISE. The first of those documents, The MDA As-Is Architecture, was completed in June 2009. The second and third documents were the MDA To-Be Architecture and the Architecture Implementation Plan respectively. In October 2010, the National MDA governing body, the Executive Steering Committee[3], authorized combining the final two documents into a single document: The National Maritime Domain Awareness Architecture Plan.

The goal of this document is to define the features, processes and technologies required to implement the Maritime Information Sharing Environment (MISE). The MISE is the main body of the Architecture Plan and is the term that embodies the participants and functions described herein.

2. Overview

The National Maritime Domain Awareness Architecture Plan (also referred to as the Architecture Plan) describes how maritime information providers may share and protect their information within the MISE. The Global Maritime Community of Interest (GMCOI) is broad and includes federal, state, local, tribal, and international partners. Maritime information must be integrated and shared across the GMCOI to gain an effective understanding of the maritime domain. The Architecture Plan is a solution to mitigate and manage gaps between the different information sharing systems and user requirements.

Being mindful of the principles identified in the National Strategy, the objectives in the MDA CONOPS and with full understanding that each participating agency has its own operational constraints; the MISE defines an operational and technical framework that enables information sharing by leveraging existing programs and systems to the greatest extent possible. Agencies and organizations with maritime interests have their own requirements, authorities, information infrastructures, and resources. The MISE accounts for each of those constraints and defines a

[3] The Executive Steering Committee members are Senior Executive Service or Flag Officers from the Departments of Defense, Homeland Security and Transportation and the Office of the Director of National Intelligence.

service oriented architectural approach that allows participation while protecting individual information and resources. Data and analytical products are shared via common data standards with access controls that enable data providers to manage sharing as defined by their respective authorities or regulations.

To meet the goals and objectives described in this plan and outlined in the National Strategy and MDA CONOPS, the MISE is comprised of four functional parts. They include:

- Trusted systems and their users
- NIEM Maritime data standards
- Common attributes for Access Control
- Information Sharing Infrastructure (ISI)

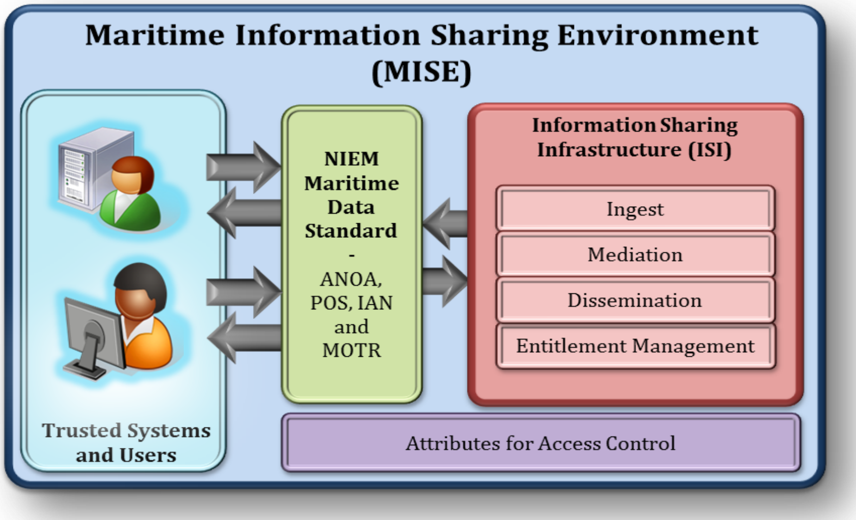

Maritime Information Sharing Environment (MISE)
Trusted Systems and Users
NIEM Maritime Data Standard - ANOA, POS, IAN and MOTR
Information Sharing Infrastructure (ISI)
Ingest
Mediation
Dissemination
Entitlement Management
Attributes for Access Control

Figure 1: Operational view of the MISE

The breadth of all possible maritime information sharing is greater than can be described in a single document or undertaken in a single effort. To scope the initial effort and manage the volume of information contained in this plan, the four focus areas were identified for the initial effort. They include:

1. Advance Notice of Arrival (ANOA). A 96-hour advance notice that all vessels inbound to US ports is required to submit which lists vessel, crew, passenger, and cargo information.
2. Indicators and Notifications (IAN). Indicators are information used to inform or contribute to an analytical process. Notifications include warnings of a possible event and alerts about the execution of an event.
3. Positions (POS). A geospatial position, course, heading, speed, and status of a vessel at a given time. A series of position reports can be combined to produce track information.
4. Maritime Operational Threat Response (MOTR). A cross-departmental conference convened as necessary to coordinated response to a maritime threat.

To ensure proper data security and entitlement management, the Architecture Plan employs an attributes-based sharing policy that defines an Information Access Policy (IAP) process. The IAP is a predefined set of rules, represented using common attributes, that governs which users can access what information. The attributes-based sharing policy assures information providers that information is properly handled and access is only granted authorized users.

3. Understanding Information Sharing

Information sharing is *not* about moving information between data providers and data consumers. It's *not* about defining common standards or security protocols. It's *not* about sharing for the sake of sharing. Information sharing *is* about improving our business processes to execute our defined missions to the best extent possible. To improve our business processes and better execute our missions, we need to gather as much relevant information as possible or provide information products that are the result of our analytical missions.

The Architecture Plan describes a repeatable information sharing process governed by common rules for information management. The Architecture Plan uses a common vocabulary to describe the information that is shared and why we're sharing it. "Why" is the business, mission or operational reason we expend resources to implement the sharing services. The "why" may be the most important but most overlooked part of any information sharing effort.

Finally, information sharing is about more than gaining access to large amounts of unprocessed or raw data. It's about defining and sharing the products of data processing, analysis or fusion. The Architecture Plan amplifies the idea that information sharing is not about discrete data elements or sharing raw sensor based data. Information sharing must move beyond sharing individual data elements and include the idea of *data products* that are the result of analytical processes or data processing capabilities and provide operational relevance. While a contact from a radar or acoustical sensor may be of value, that same contact correlated or fused to other data providing vessel name and characteristics is more valuable to the community.

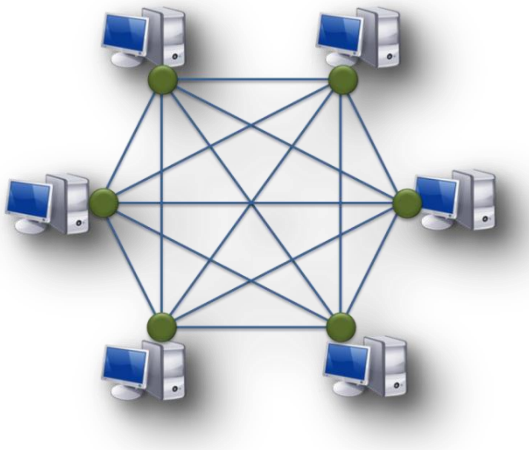


Figure 2: Point-to-point sharing

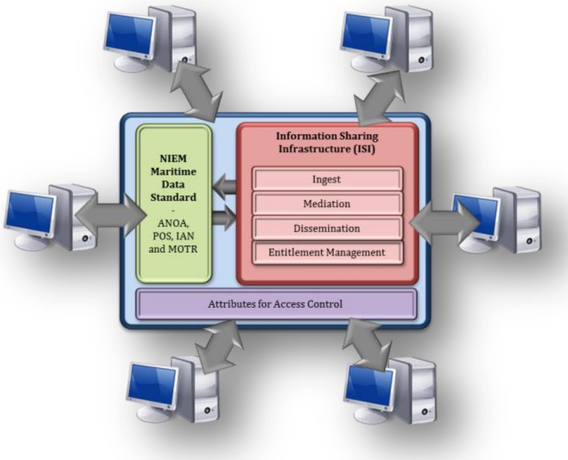


Figure 3: MISE services based sharing

Figure 2 represents the complexity of current point-to-point sharing activities. Each participating system must establish a discrete connection point to every other system. This type of sharing environment requires every participant to update its connection point whenever a new partner joins the environment. Figure 3 represents MISE services based sharing. Each participating system maintains only one connection point to the MISE. When a new participant joins the MISE, no changes are required by existing members. The ISI continues entitlement management for the new and existing information according to the common security attributes.

3.1. INFORMATION SHARING BUSINESS PROCESS

Successful information sharing activities are the result of operational, information and technological understanding achieved through a well-defined and routinely implemented process. The Architecture Plan describes a multistep information sharing process.

1. Describe the operational use case being supported by the information sharing.
2. Identify the specific data elements required to support the use case.
3. Develop a standard definition, model or product for the information to be shared.
4. Identify any legislative or policy driven constraints on the information.
5. Implement appropriate controls to ensure proper entitlement management.
6. Implement and monitor the sharing service.

Each of the six steps will be discussed in greater detail throughout the Architecture Plan.

3.2. NATIONAL DATA PRODUCTS: A NATIONAL ASSET

The first of the core principles outlined in the National Strategy is *Information as a National Asset*. While information must be safeguarded, it first must be made available. Each data provider has legal, contractual or policy reasons that prevent all the data they maintain or process from being shared. However; the *products* of their analytical and management process may have broader releasability. It's the obligation of those with data products to make those products discoverable and sharable; even if that means the release of multiple versions with different security controls. Leveraging the security markings, such as the attributes in NIEM-M based exchanges, a single data model can be used to support multiple versions of a product. The idea of *Information as a National Asset* can be realized through national information products. Information products, derived from National maritime programs like the Department of Transportation's Maritime Administration (MARAD) or the Department of Homeland Security Maritime Information Global Network (MAGNet), make the results of complex analysis or data fusion processes available to consumers who might have less mature analytical or IT processes. The availability of information products from these types of programs relieve other Agencies or programs from duplicating the analysis thus providing cost saving and resulting in a common maritime understanding. Products like *Levels of Awareness*[4] can be provided via a simple Google Earth interface yet give the consumer a deep understanding of the maritime domain.

[4] *Level of Awareness* (LOA), as defined by the Department of Defense, is a standard definition of maritime understanding to support national security. Details about LOA can be found in the LOA Information Exchange Package Documentation (IEPD) attachment.

3.3. USE CASES

The first step to any information sharing effort is the definition of the operational context; the reason why the sharing action should even take place. Broad ideas like "improve national security" or "stop illegal activities" are reasons that have been used to define the "why" of information sharing but rarely provide the fidelity required to define the specific data elements that must be shared. These broad descriptions lead to long and often unproductive discussions about what specific data should be shared, who has the authority to provide or consume the information, and what it will be used for once it is shared. To ensure both data providers and consumers are able to work within their own legislative or policy bounds, the "why" of the sharing must be as specific as the data elements that are shared.

Use cases have been defined for each of the initial focus areas within the Architecture Plan. The use cases were used to identify the specific data elements contained in the supporting model and how the information would be shared. In lieu of listing the entire set of use cases, they are represented as graphic (Figure 5) depicting common features, functions and data elements.

3.4. DATA STANDARDS

Within the Architecture Plan, "data standards" are the common vocabulary used to describe the shared data objects and the models themselves. The National Information Exchange Model[5] (NIEM) and NIEM-Maritime serve as the reference library used to develop the specific data objects to be shared. The information contained in NIEM is stored in an eXtensible Markup Language (.xml) format. The technical representation of .xml is complex and it can be difficult to track the relationship of NIEM and NIEM-Maritime components. To facilitate discussion and demonstrations with business and process managers, a graphical representation of each data model has been developed.

Figure 4 is a graphical representation of the logical data model for Position data. Each Position record is defined with *Position* and *Vessel Information* components as defined in NIEM. *Record Metadata* provides security and record control information.

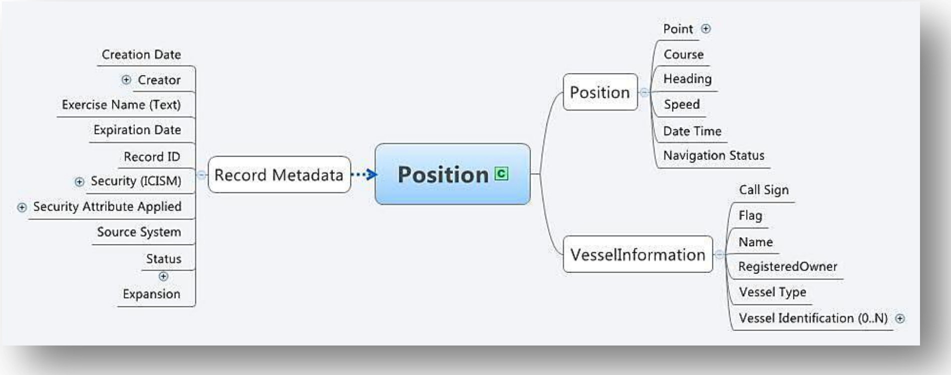


Figure 4: Position data model

[5] NIEM, a Department of Justice and Department of Homeland Security led effort, is the National guideline for sharing data between United States government agencies and their partners.

Figure 5 is a graphical representation of the request use case for position data. Positions can be requested by *GeoLocation* or *Vessel*, both with specific selection and bounding variables, and filtered by *Record Metadata*. Examples of request strings are available in the *Implementation Guide* (Appendix A)

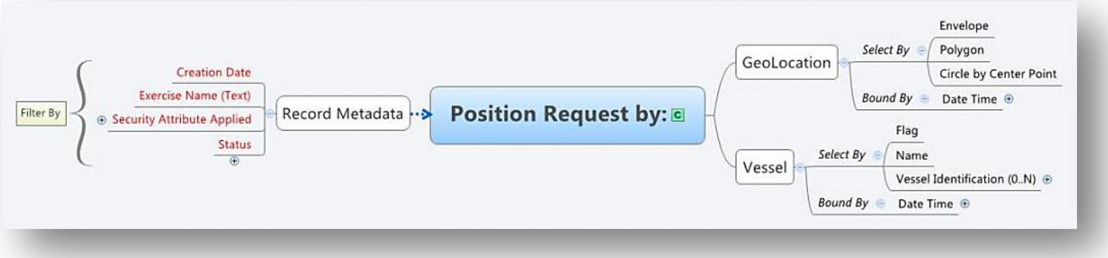


Figure 5: Position Request Use Cases

3.5. SECURITY ATTRIBUTES

Security attributes and record metadata are the elements used to provide access control for the data object. Security attributes are applied at the message, record and user level.

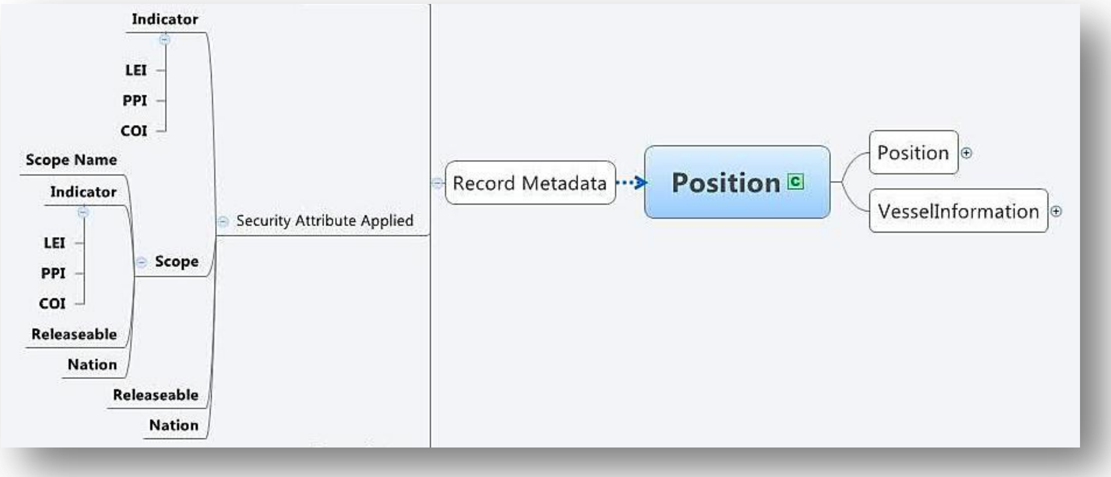


Figure 6: Record level security attributes

As shown in figure 6, security attributes include Indicator, Releasable, Nation and Scope. Currently, Indicators are Law Enforcement (LEI), Protected (PPI) and the entire community (COI) with COI being all participants of the MISE. Releasable indicates if the record is publicly releasable and Nation is a list of nations that can receive the record. Scope, which is made up of a scope name and all the previous listed attributes is designed to allow further control of the record. Security attributes are separate from, but an integral part of, the NIEM–Maritime data model. Full details about *Security Attribute Specifications* and their implementation can be found in Appendix B.

4. The MISE

The Maritime Information Sharing Environment, MISE, is the main body of the Architecture Plan. MISE is the term that embodies the participants and functions described in the Architecture Plan. The MISE consists of:

- Trusted systems and their users
- Information Sharing Infrastructure
- NIEM data standards
- Security Attributes for Entitlement Management

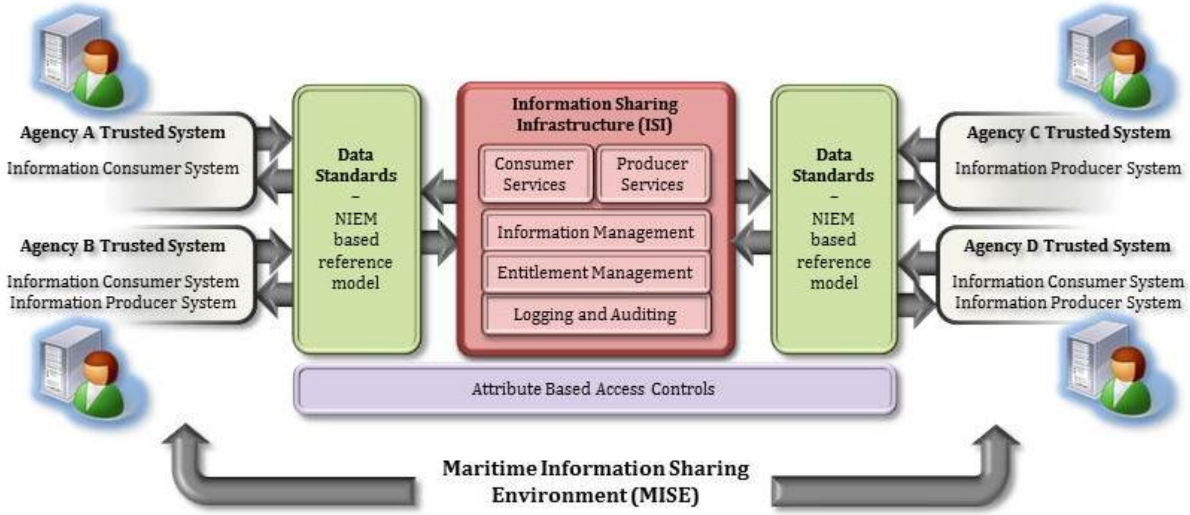
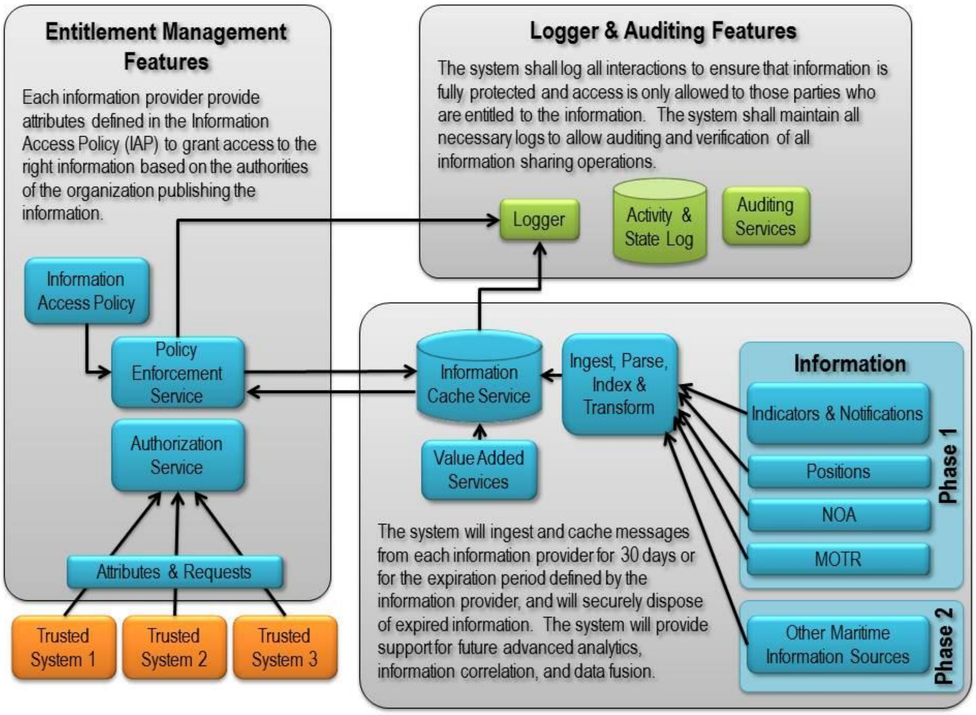

Agency A Trusted System
Information Consumer System
Agency B Trusted System
Information Consumer System
Information Producer System
Data Standards
-
NIEM based reference model
Information Sharing Infrastructure (ISI)
Consumer Services
Producer Services
Information Management
Entitlement Management
Logging and Auditing
Data Standards
-
NIEM based reference model
Agency C Trusted System
Information Producer System
Agency D Trusted System
Information Consumer System
Information Producer System
Attribute Based Access Controls
Maritime Information Sharing Environment (MISE)

Figure 7: Maritime Information Sharing Environment

4.1. Information Sharing Infrastructure (ISI) Features

The Information Sharing Infrastructure (ISI) is the central hub of the MISE and the component that provides services to the trusted systems. Participating agencies will be associated to the MISE via their trusted system which will allow them to share or retrieve information by connecting and interacting with the ISI. Trusted systems can serve as information providers, consumers, or both. Trusted systems do not interact directly with, or access, other trusted systems; rather they interact only through the ISI. A trusted system is a participant's new or legacy IT system that allows a user to process, manage or view available maritime information. A full description of trusted system requirements is outlined in Appendix F.

At a high level, the ISI features are captured in three functional areas: Entitlement Management, Information Management, and Logging/Auditing. The figure below depicts the information flow within the ISI, and how individual features are broken out across each functional area.


Entitlement Management Features
Information Access Policy
Policy Enforcement Service
Authorization Service
Attributes & Requests
Trusted System 1
Trusted System 2
Trusted System 3
Logger & Auditing Features
Logger
Activity & State Log
Auditing Services
Information Cache Service
Ingest, Parse, Index & Transform
Value Added Services
Information
Indicators & Notifications
Positions
NOA
MOTR
Phase 1
Other Maritime Information Sources
Phase 2

Each information provider provide attributes defined in the Information Access Policy (IAP) to grant access to the right information based on the authorities of the organization publishing the information.

The system shall log all interactions to ensure that information is fully protected and access is only allowed to those parties who are entitled to the information. The system shall maintain all necessary logs to allow auditing and verification of all information sharing operations.

The system will ingest and cache messages from each information provider for 30 days or for the expiration period defined by the information provider, and will securely dispose of expired information. The system will provide support for future advanced analytics, information correlation, and data fusion.

Figure 8: Information Sharing Infrastructure Features

A full description of the ISI is contained in the appendices.

4.2. MISE Framework

The Architecture Framework, described in detail in the *Framework* section is comprised of four architectural views: Data, Services, Security, and Technical Operations as depicted in the following figure.

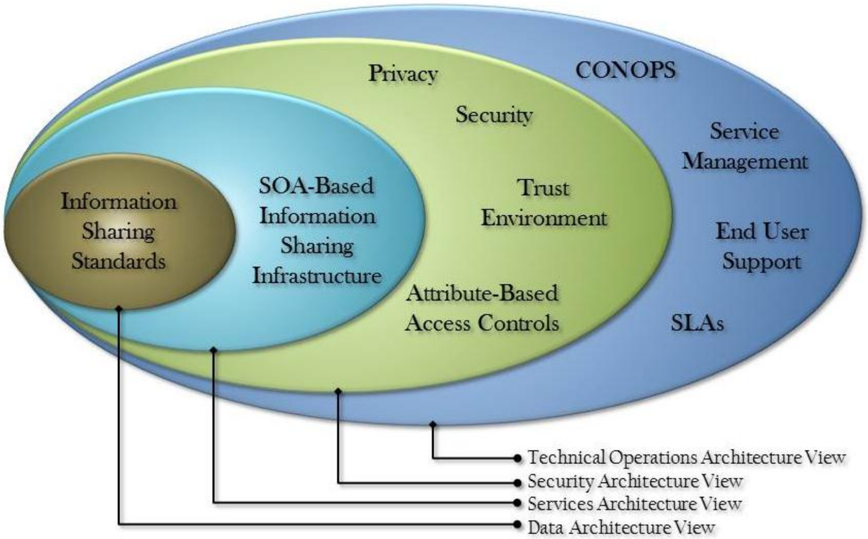

Information Sharing Standards
SOA-Based Information Sharing Infrastructure
Privacy
Security
Trust Environment
Attribute-Based Access Controls
CONOPS
Service Management
End User Support
SLAs
Technical Operations Architecture View
Security Architecture View
Services Architecture View
Data Architecture View

Figure 9: The Four Architectural Views of the MISE Framework

Data Architecture View

The Data Architecture View is the center of the architecture framework. It includes the standard characteristics and vocabulary used to define standards for information sharing. These standards are a part of the NIEM.

Services Architecture View

The Services Architecture View builds upon the Data Architecture View, and is the foundation for the MISE. It includes the information sharing infrastructure that provides common service interfaces for trusted systems to share information.

Security Architecture View

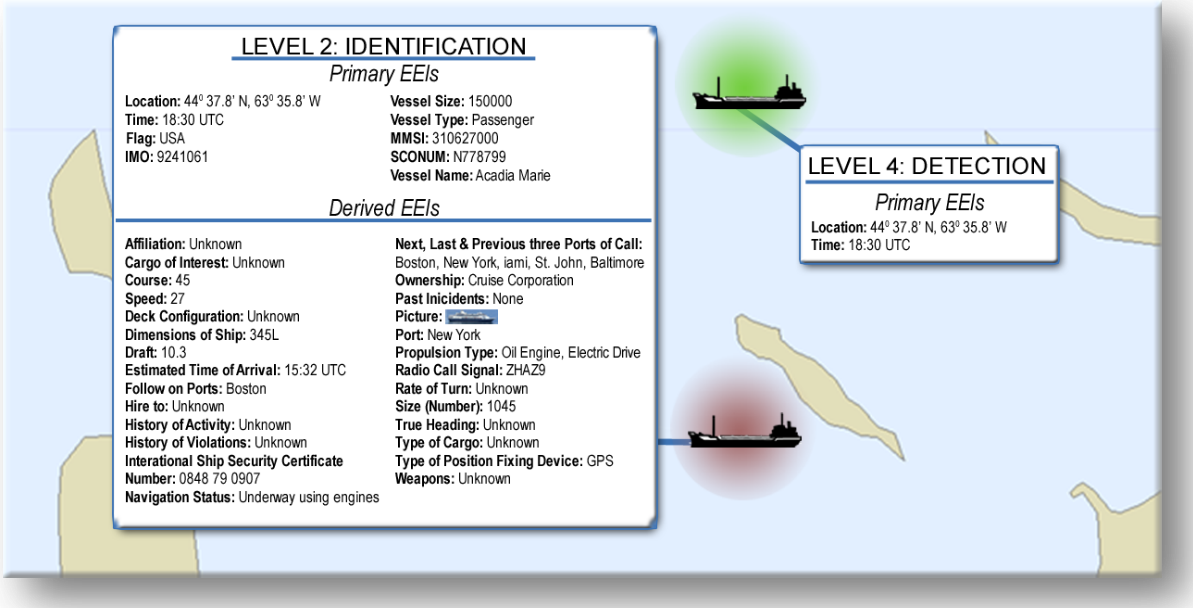
The Security Architecture View ensures that the shared information is protected. It includes the common security attributes and the association of the trusted systems.

Technical Operations View

The Technical Operations View defines how the framework as a whole is managed and operated. It includes governance and agreements used to manage the MISE.

Below is an example of how MISE services are used together across all four architectural views of the framework to deliver value and insight to users.

In a given region surrounding a port, a graphical display shows incoming and outgoing vessels. Users of the system can click on a vessel and retrieve detailed information about the vessel. Users also have the ability to display tracks showing the path taken by the vessel to reach its current position. The notional graphic below shows how a simple interface can be used to receive large amounts of data from the MISE. Representative Level of Awareness information is shown.



LEVEL 2: IDENTIFICATION

Primary EEIs

Location: 44^0 37.8' N, 63^0 35.8' W
Time: 18:30 UTC
Flag: USA
IMO: 9241061

Vessel Size: 150000
Vessel Type: Passenger
MMSI: 310627000
SCONUM: N778799
Vessel Name: Acadia Marie

Derived EEIs

Affiliation: Unknown
Cargo of Interest: Unknown
Course: 45
Speed: 27
Deck Configuration: Unknown
Dimensions of Ship: 345L
Draft: 10.3
Estimated Time of Arrival: 15:32 UTC
Follow on Ports: Boston
Hire to: Unknown
History of Activity: Unknown
History of Violations: Unknown
Interational Ship Security Certificate Number: 0848 79 0907
Navigation Status: Underway using engines

Next, Last & Previous three Ports of Call: Boston, New York, iami, St. John, Baltimore
Ownership: Cruise Corporation
Past Inicidents: None
Picture:
Port: New York
Propulsion Type: Oil Engine, Electric Drive
Radio Call Signal: ZHAZ9
Rate of Turn: Unknown
Size (Number): 1045
True Heading: Unknown
Type of Cargo: Unknown
Type of Position Fixing Device: GPS
Weapons: Unknown

LEVEL 4: DETECTION

Primary EEIs

Location: 44^0 37.8' N, 63^0 35.8' W
Time: 18:30 UTC

Figure 10: Representational of MISE Services

4.3. MISE EVOLUTION

The Architecture Plan, including the implementation of the ISI and definition of data standards and processes, is partially bounded by the technical and process maturity of the participating maritime community. As the community and its processes mature, the trust of data providers and their products will grow. Improvements in the level of trust between the MISE participants and technical capabilities will encourage more sharing and additional services. With time, the MISE will provide greater services and value added products. As that happens, greater fidelity in the security attributes will be required. The MISE has been designed and developed to deliver the needs of today's user community while allowing the flexibility to grow with the maturity of that community.

5. Security

To many participants of the MISE, security is the most import aspect of implementation. For the Architecture Plan, security is designed to be implemented the same across the MISE yet controlled by individual data providers.

Security is based upon a common set of attributes. The three categories for attributes are:

1. Entity Attributes: Attributes associated to a trusted system
2. User Attributes: Attributes associated to a specific user
3. Data Attributes: Attributes associated to data record

When a data provider publishes information, they will provide the appropriate security attributes to the ISI for each record. When a trusted system requests information on behalf of a user, the ISI will ensure the attributes of the user match those of the data provider and only share the appropriate marked information. Since the attributes are asserted at the time the information is published, two direct benefits are achieved:

1. The attributes are immediately applied to the data. No additional system changes are required.
2. The attributes are associated to the information and passed to data consumers. This ensures the data consumer understands the protection and management requirements of the data provider.

5.1. SECURITY OVERVIEW

In order for users to feel comfortable that their information is protected, the Architecture Plan defines the MISE trust fabric. The MISE trust fabric is an .xml document that is digitally signed by the MISE Certificate Authority (CA). The trust fabric is a critical document that describes each trusted system in Security Assertion Markup Language (SAML) metadata format. A sample of the trust fabric and SAML assertions is contained in Appendix C. From a security perspective, the value of the trust fabric is that it is an electronically signed document that includes the certificates of all trusted systems participating in the MISE. The electronic signing of the trust fabric ensures tampering can be detected and the certificates in the document can be trusted. The certificates are used to establish SSL connections for all other communications.

Once a trusted system has been added to the trust fabric, every connection to the ISI, to publish or consume information, includes information about the trusted system and the user system it's representing. At multiple points during the transaction, the ISI validates the request, the attributes and the data to ensure only the correct information is shared.

Figure 11 shows the steps in a typical transaction. Each time the request or the information touches the "ISI: Policy Enforcement Service" bar, the attributes of the trusted system and the user are compared against the security attributes assigned to the information by the data provider to ensure the transaction is allowed. Throughout the process, every step and decision is entered in the audit log.

Color Legend	
🟧	Information Consumer
🟪	Information Provider
🟦	ISI
🟩	Logger

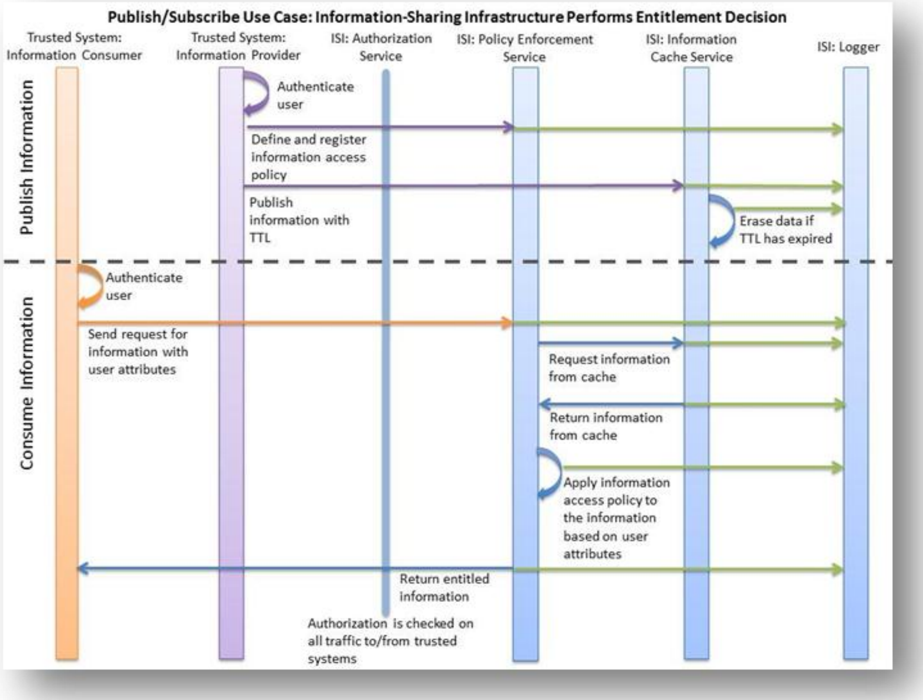

Publish/Subscribe Use Case: Information-Sharing Infrastructure Performs Entitlement Decision
Trusted System: Information Consumer
Trusted System: Information Provider
ISI: Authorization Service
ISI: Policy Enforcement Service
ISI: Information Cache Service
ISI: Logger
Publish Information
Authenticate user
Define and register information access policy
Publish information with TTL
Erase data if TTL has expired
Consume Information
Authenticate user
Send request for information with user attributes
Request information from cache
Return information from cache
Apply information access policy to the information based on user attributes
Return entitled information
Authorization is checked on all traffic to/from trusted systems

Figure 11: Information Sharing Sequence Diagram

6. Governance

Successful information sharing can only be achieved if information providers can be assured that information is protected in accordance with applicable doctrine. The MISE governance is comprised of three major areas:

1. Service Management
   a. Service Level Agreements
2. Organizational Roles and Responsibilities
   a. MISE Board of Directors
   b. MISE Management

c. Trusted Systems

3. End User Support

a. Three Tiers – Local, Trusted System, MISE

Full detail of the proposed MISE governance is contained in Appendix F.

The below figure depicts the major areas of the MISE governance.

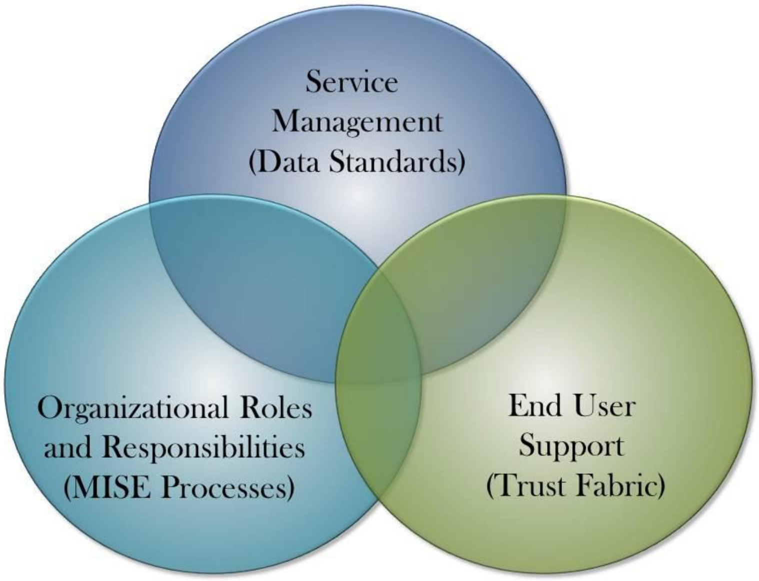

Service Management (Data Standards)
Organizational Roles and Responsibilities (MISE Processes)
End User Support (Trust Fabric)

Figure 12: MISE Governance

6.1. SERVICE MANAGEMENT

To correctly govern the information sharing process, the MISE and each trusted system will need to have service level agreements (SLA) which formally define the level of service that should be expected from each. The development and management of MISE SLAs will be the responsibility of the MISE Management Organization. The service level agreements will define:

- The type of services that will be used for the information exchange
- Terms for selecting, accepting, maintaining and transitioning services into the MISE
- Terms for executing and evaluating the quality of service of information exchanges
- Terms for executing the service agreement

Term for executing the service agreement will include:

- Contacts and Role assignment - Responsible organizations and roles that have access to the trusted systems and the MISE
- Reporting - Logs will be provided based on terms laid out in the SLA
- Review Process - Quarterly reviews will be established that include discussion on SLA fulfillment and future projects that may affect the SLA

- Performance Level Guidelines - Outlines the inbound and outbound information flows to and from the ISI and expected behavior from each trusted system and the MISE
- Uptime Requirements - Outlines the expected availability that the trusted systems must deliver as well as the expected availability that it will provide
- Equipment Support Requirements - Access and security of the trusted systems and the MISE
- Third Party Sharing – Can data be shared beyond the receiving trusted system other than as defined in the releaseability attribute?

6.2. ORGANIZATIONAL ROLES AND RESPONSIBILITIES

The roles and responsibilities for the execution of the MISE are divided between three organizational groups.

MISE Board of Directors

The MISE Board of Directors (BoD) is the executive-level body with representation from primary stakeholders that guides the MISE and is the final authoritative body to make decisions for the environment.

MISE Management Organization

The MISE Management Organization is responsible for the day-to-day operations for the MISE and will facilitate the identification, collaboration, management, movement, and processing of maritime information by leveraging the MISE. The MISE Management Organization is also responsible for the establishment and management of the Configuration Control Board.

Trusted Systems

Trusted systems, both Information Providers / Information Consumers, are responsible for adhering to the information sharing environment rules established by MISE management.

6.3. END USER SUPPORT

The MISE help desk structure focuses on solving problems as close to the user as possible. The MISE Technical Authority serves as the primary point of contact for the MISE Help Desk, but complex technical issues across the environment will be handled according to the following three-tier structure:

Tier 1: Local Help Desk Support

All issues encountered by users should be first reported to the users local help desk. This level of user assistance is provided by the user's local department to resolve simple issues reported by users.

Tier 2: Trusted System Help Desk Support

Any issue the local help desk cannot resolve will be elevated to the Tier 2: Trusted System Help Desk Support level. The trusted system help desk should attempt to resolve the issue, and will contact the MISE help desk support staff if the issue relates to an information provider or the ISI.

Tier 3: MISE Help Desk Support

Any issue that cannot be resolved at the Tier 2 level should be escalated to the Tier 3: MISE Help Desk. Issues that are resolved at this level will be tracked in the MISE Technical Authority issue tracker database.

7. Implementation

Information sharing has become the common thread to success and mission accomplishment for agencies and organizations across all levels of government (federal, state, local, and tribal) as well as the private sector. While it's understood information sharing benefits National Security, sharing efforts continue to be impeded by technical, legal, or policy concerns. The Architecture Plan defines a low cost, technical solution that is easily implemented. The Architecture Plan also puts mechanisms in place to manage legal and policy concerns. The uniqueness and benefit of the Architecture Plan is that it allows agencies and organizations to utilize their own requirements, authorities, information infrastructures, and resources.

It is not feasible to develop an environment to accommodate specific concerns of every agency; however, the MISE supports sharing and management of the majority of unclassified maritime information. Having this common central hub for unclassified maritime information that the Nation collects and processes will support and improve National Maritime Security.

7.1. Impact & Scope

The MDA CONOPS and the National Strategy speak to all levels of partnership sharing, public, private, domestic and international. The concepts contained in the Architecture Plan have been reviewed and adopted within the U.S. Government and is being shared with other domestic partners. The NIEM-Maritime standard has been reviewed and is being adopted by domestic *and* international partners. The impact and scope of the Architecture Plan supports that core principals, goals and objectives in the MDA CONOPS and the National Strategy.

7.2. Trusted System IT Requirements

Becoming a trusted system is the first and most important step to becoming part of the MISE. To be considered for a trusted system and entered into the trust fabric, as system must:

1. Have a user management system. The system must manage anticipated and unanticipated (registered and unregistered) users. The system must be able to provide attributes applicable to every user.
2. Provide for attribute management through the trusted system if third party sharing is possible.
3. Be able to implement restful web services that produce and consume NIEM compatible messages.
4. Demonstrate the need to have access the information contained in the MISE.
5. The approval of the MISE Governance Board.

Full details about becoming a trusted system are contained in the appendices.

7.3. RESOURCE REQUIREMENTS

The "cost" of the Architecture Plan and its resourcing will continue to be a discussion point within the interagency community. Ultimately, the MDA Executive Steering Committee will determine who is responsible for the various parts of the MISE. It will however be the responsibility of the departments, agencies or offices that own and maintain the trusted systems to ensure those systems are properly resourced and maintained.

THIS PAGE INTENTIONALLY LEFT BLANK

FRAMEWORK

1. Overview

The National Concept of Operations for Maritime Domain Awareness (MDA) describes an information sharing end-state in which an authorized user is able to access the right information at the right time. Access is managed through entitlements based on the needs, rights and authorities of both the user consuming the information and of the organizations providing the information. Although a key MDA tenet it to provide the broadest information sharing possible, it can only be achieved if information providers are assured the information is protected according to their requirements. The establishment of the Maritime Information Sharing Environment (MISE) and the data standards as described in this plan is the first step toward that assurance.

The following figure depicts the Operational Overview (OV-1) for an information sharing environment as a national capability to share and search maritime information across organizational boundaries. Within the information sharing environment depicted in OV-1, an infrastructure exists to effectively link participants and information. This results in a distributed, protected, and trusted environment.

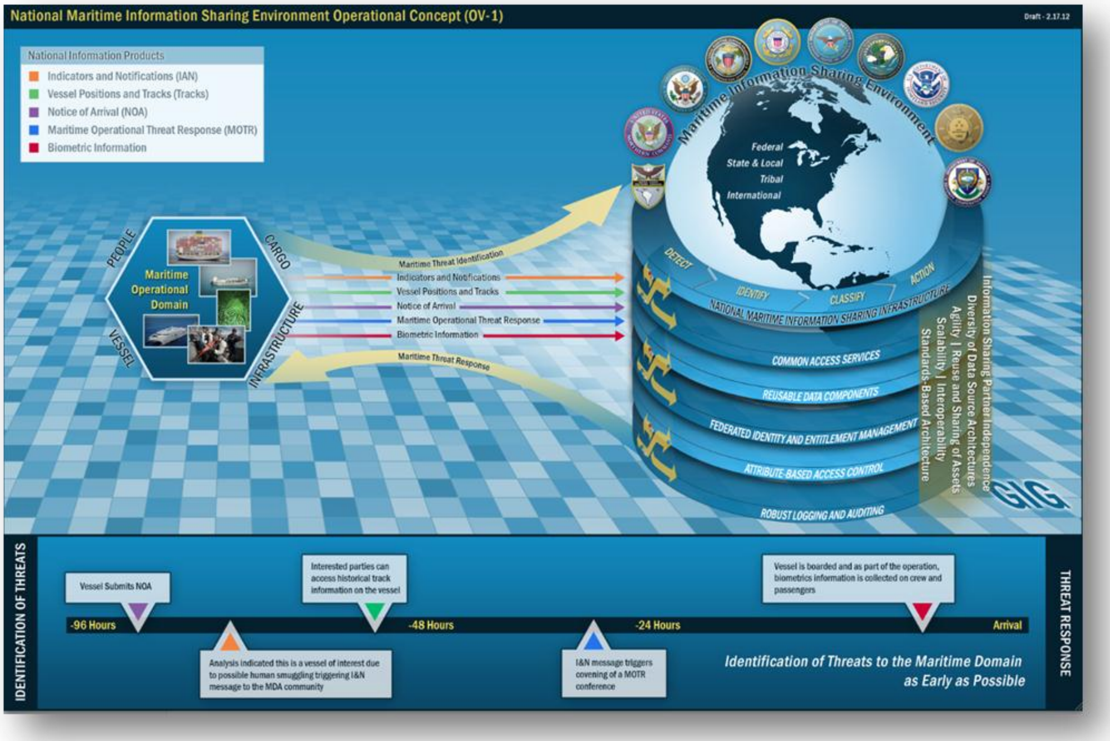

National Maritime Information Sharing Environment Operational Concept (OV-1)
Draft - 2.17.12
National Information Products
🟧 Indicators and Notifications (IAN)
🟩 Vessel Positions and Tracks (Tracks)
🟪 Notice of Arrival (NOA)
🟦 Maritime Operational Threat Response (MOTR)
🟥 Biometric Information
PEOPLE
CARGO
VESSEL
INFRASTRUCTURE
Maritime Operational Domain
Maritime Threat Identification
Indicators and Notifications
Vessel Positions and Tracks
Notice of Arrival
Maritime Operational Threat Response
Biometric Information
Maritime Threat Response
Maritime Information Sharing Environment
Federal
State & Local
Tribal
International
DETECT
IDENTIFY
CLASSIFY
ACTION
NATIONAL MARITIME INFORMATION SHARING INFRASTRUCTURE
COMMON ACCESS SERVICES
REUSABLE DATA COMPONENTS
FEDERATED IDENTITY AND ENTITLEMENT MANAGEMENT
ATTRIBUTE-BASED ACCESS CONTROL
ROBUST LOGGING AND AUDITING
Information Sharing Partner Independence
Diversity of Data Source Architectures
Agility | Reuse and Sharing of Assets
Scalability | Interoperability
Standards-Based Architecture
GIG
IDENTIFICATION OF THREATS
Vessel Submits NOA
Interested parties can access historical track information on the vessel
Vessel is boarded and as part of the operation, biometrics information is collected on crew and passengers
-96 Hours
-48 Hours
-24 Hours
Arrival
Analysis indicated this is a vessel of interest due to possible human smuggling triggering I&N message to the MDA community
I&N message triggers covening of a MOTR conference
Identification of Threats to the Maritime Domain as Early as Possible
THREAT RESPONSE

Figure 13: MISE Operational View

The Operational Overview identifies four subject areas that make up the Maritime Operational Domain: Cargo, Infrastructure, Vessels, and People. Each of the participants in the Global Maritime Community of Interest (GMCOI) contributes to the National MDA in terms of these four subject areas. Specifically, the GMCOI interact regarding the monitoring and collection of information, as well as fusion and analysis of data. The MISE focuses initial efforts on securely sharing the following *National* information products within the GMCOI:

1. Advance Notice of Arrival (NOA) - A 96-hour advance notice that vessels inbound to US ports are required to submit, which lists vessel, crew, passenger, and cargo information.
2. Indicators and Notifications (IAN) - , including warnings and alerts:
    a. Indicator – Any piece of discreet information, processed or otherwise that is used to inform or contribute to an analytical process
    b. Notification – Directed communication
        i. Warning – Notification of a possible activity or event
        ii. Alert – Notification of the actual or eminent execution of an event.
3. Positions (POS) - A geospatial position, course, heading, speed, and status of a vessel at a given time. A series of position reports can be combined to produce track information
4. Maritime Operation Threat Response (MOTR) - A cross-departmental conference convened as necessary to coordinate response to a maritime threat.

2. Objective

The overarching goal of the Architecture Plan is to establish information sharing process and framework to advance information sharing capabilities. The objective is for agencies and organizations within the GMCOI to be able to utilize their own requirements, authorities, information infrastructures, and resources to share information within the community that would assist operational missions. The MISEsimply provides an environment for the successful transfer of information between information consumers and providers.

This section provides technical managers with greater detail for the technical implementation of the architecture. Documents and processes that are required to conduct specific actions within the architecture are referenced, but not explained in detail.

3. Definitions

The definitions below are used in the following sections to describe the participants and processes within the MISE.

Attributes	Characteristics of a persona that defines a user in a particular role (sent by a trusted system to the ISI).
Decision Authority	The authoritative role to enforce information access policy based on user attributes to determine what information will be provided to a user.

Entitlement(s)	The result of determining access by evaluating IAP against user attributes.
Identity Provider	A service provider that creates, maintains, and manages identity information and provides authentication services to other systems.
Information Access Policy (IAP)	A rule set that defines the attributes users must possess to allow access to information.
Information Consumer	A type of trusted system that authenticates users, using an internal or external identity provider, and passes information access requests and user attributes on behalf of the user to the ISI.
Information Provider	A type of trusted system that provides both maritime information and corresponding security attributes to the ISI.
Information-Sharing Infrastructure (ISI)	Handles requests from the trusted systems on behalf of the users. Operating on the user attributes, it will make entitlement decisions, processing messages from the information providers based on their security attributes and providing responses to the trusted system.
Request Broker	A role of the ISI to forward requests for information from the information consumer to the information provider.
Service Provider	An entity that provides services to other entities.
Trusted System	A system that is recognized and authenticated by the ISI and operates in a role of information provider and/or information consumer. It also passes information or information access requests and user attributes to the ISI. The trusted system relies on its own internal processes to authenticate users.
User	Any person, organization, system, etc. that authenticates via the trusted system in order to access the ISI.

Table 1: MISE Term Definitions

4. Information Sharing Process

Data movement within the MISE is between trusted systems and the ISI. Central to the information sharing process are two points. First, the decision authority either rests within the ISI or the information provider. That is, either the ISI or the information provider compares the IAP with the user attributes and decides what information should be provided back to the user. Second, information will either be cached within the ISI or stored by the information provider. Caching the information, allows the ISI to quickly respond to requests and to protect the information provider systems from query overload. These two points are addressed within the following two use cases of the ISI. Note that in all use cases, the same IAP is applied.

In use case 1, the ISI caches the information *and* acts as the decision authority.

In use case 2, the ISI acts as a request broker, passing both the request and the user attributes from the information consumer system to the information provider system. The information provider system acts as the decision authority.

The following table summarizes roles with respect to each of the use cases. The distinguishing characteristic between the use cases pertain to which system has decision authority and whether or not caching is employed.

Function	Use Case 1	Use Case 2
Caching	ISI	N/A
Decision Authority	ISI	Information Provider
Logging	ISI	ISI
Request Broker	N/A	ISI
Trusted System Authentication	ISI	ISI
User Authentication	Trusted System	Trusted System

Table 2: Roles Across Each Use Case

4.1. USE CASE 1: ISI ACTS AS DECISION AUTHORITY AND PERFORMS CACHING

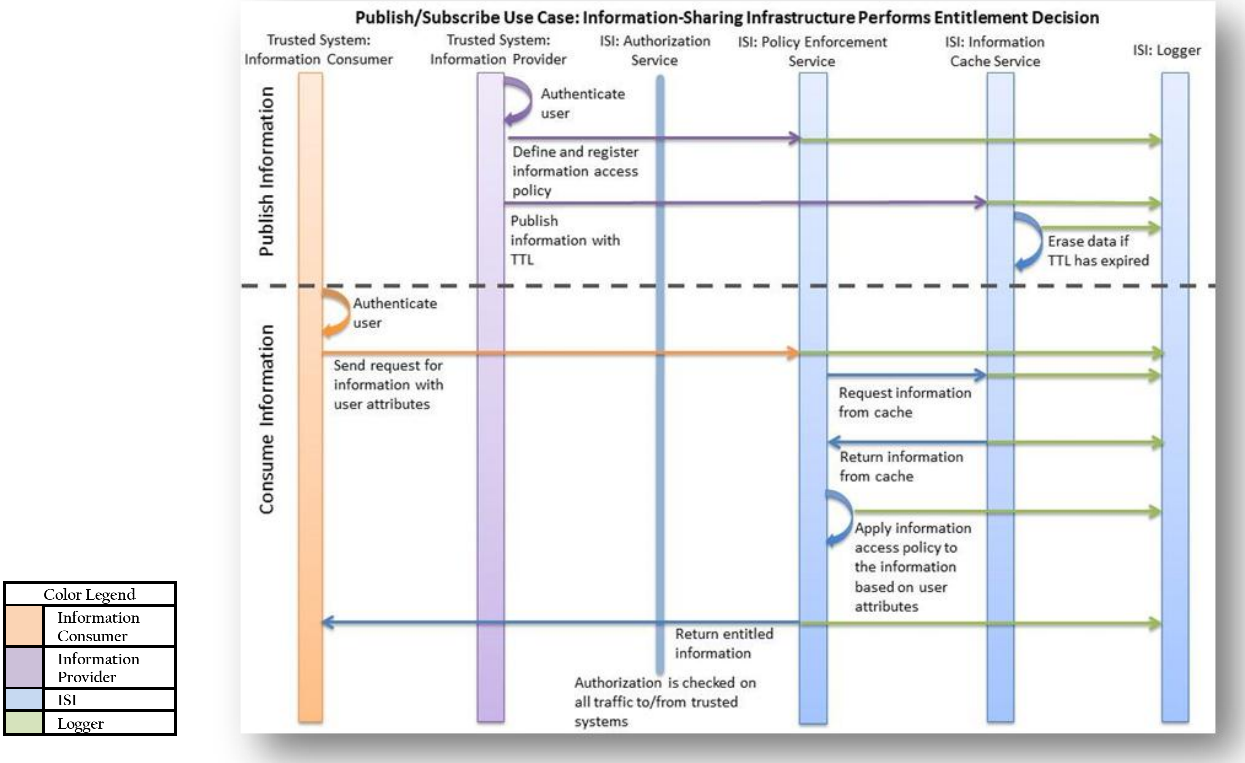

Publish/Subscribe Use Case: Information-Sharing Infrastructure Performs Entitlement Decision
Trusted System: Information Consumer
Trusted System: Information Provider
ISI: Authorization Service
ISI: Policy Enforcement Service
ISI: Information Cache Service
ISI: Logger
Publish Information
Authenticate user
Define and register information access policy
Publish information with TTL
Erase data if TTL has expired
Consume Information
Authenticate user
Send request for information with user attributes
Request information from cache
Return information from cache
Apply information access policy to the information based on user attributes
Return entitled information
Authorization is checked on all traffic to/from trusted systems
Color Legend
Information Consumer
Information Provider
ISI
Logger

Figure 14: Use Case 1 Process

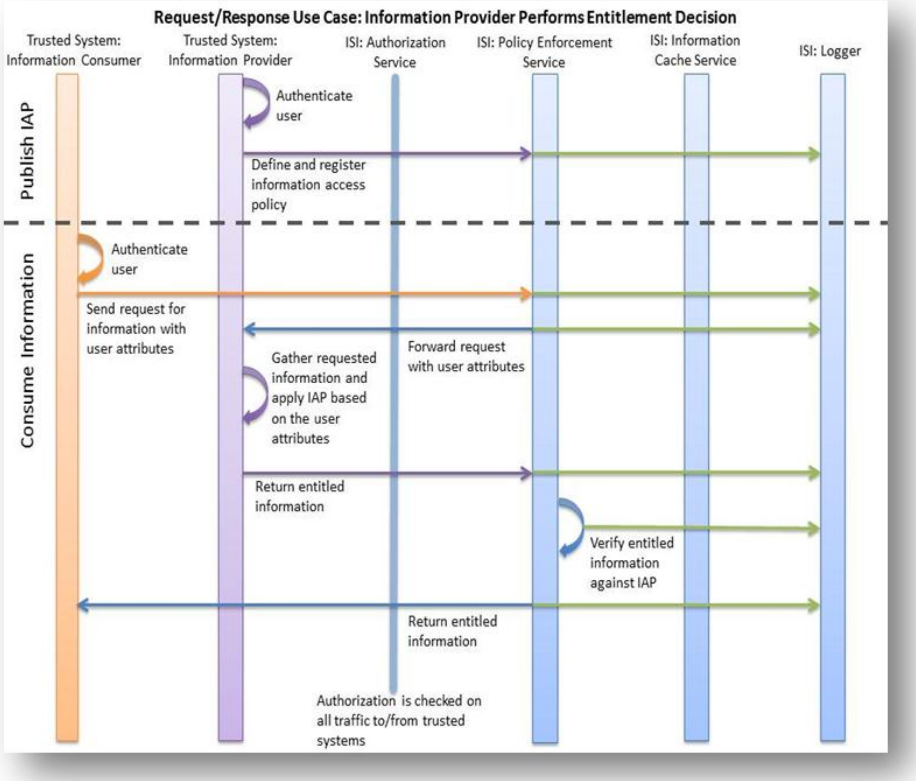
4.1.1. PUBLISH INFORMATION

A trusted system, in the role of an Information Provider, authenticates a user. The Information Provider can publish information with a Time to Live (TTL) to the ISI's information cache. This allows for messages to be published by the information provider on a constant basis and cached by the ISI to handle repeated queries from consumers for the same information. Any information in the ISI cache is purged when the TTL is reached.

4.1.2. Consume Information

A trusted system in the role of an Information Consumer authenticates a user using either an internal or external identity provider. From that trusted system, the user makes a request for information from the ISI. The trusted system supplies a set of attributes that define that user's persona along with the request to the ISI. The ISI queries its own local cache for matching information and compares the IAP attributes in each record from the data provider to the user attributes provided with the request. By evaluating the IAP against the user attributes, the ISI makes an entitlement decision. According to the IAP, the message response is tailored to match the entitlements and the resulting information is returned to the user.

4.2. Use Case 2: ISI Acts as Request Broker, Info Provider Acts as Decision Authority; No Caching


Request/Response Use Case: Information Provider Performs Entitlement Decision
Trusted System: Information Consumer
Trusted System: Information Provider
ISI: Authorization Service
ISI: Policy Enforcement Service
ISI: Information Cache Service
ISI: Logger
Publish IAP
Authenticate user
Define and register information access policy
Consume Information
Authenticate user
Send request for information with user attributes
Forward request with user attributes
Gather requested information and apply IAP based on the user attributes
Return entitled information
Verify entitled information against IAP
Return entitled information
Authorization is checked on all traffic to/from trusted systems

Color Legend	
🟧	Information Consumer
🟪	Information Provider
🟦	ISI
🟩	Logger

Figure 15: Use Case 2 Process

4.2.1. Publish Information

A trusted system in the role of an Information Provider authenticates a user. Since the trusted system is authorizing the user request and providing the requested data, no data is published to the ISI.

4.2.2. CONSUME INFORMATION

A trusted system in the role of an Information Consumer authenticates a user. From that trusted system, the user makes a request for information from the ISI. The trusted system supplies a set of attributes that define that user's persona. The ISI forwards the request with user attributes to the information provider. The information provider gathers the requested information and applies their IAP based on the user attributes. The information provider returns the entitled information back to the ISI. The ISI verifies the entitled information against the IAP and returns the resulting information to the user.

4.3. COMPARING THE USE CASES

Information providers will implement the use case that best meets their requirements. Table 2 provides a comparison of the two proposed use cases from the information provider's perspective.

Use Case	Benefits	Implications
Use Case 1 ISI Acts as Information Broker (ISI has Decision Authority and ISI Performs Caching)	• Eliminates cost of servicing requests • Reduces implementation costs associated with enforcing information access policy	• Information Provider must tag messages with security attributes to convey information controls • Information Provider must keep cached information in the ISI current
Use Case 2 ISI acts as Request Broker only (Information Provider has Decision Authority; No Caching)	• Information Provider maintains maximum control over data by assuming IAP enforcement responsibility	• Increased implementation costs associated with enforcing information access policy • Information Provider must tag messages with security attributes to convey information controls • Information Provider must ensure capacity to service requests from National Maritime Federation

Table 3: Use Case Comparison

4.4. INFORMATION SHARING INFRASTRUCTURE (ISI) FEATURES

The Information Sharing Infrastructure (ISI) is the central hub of the MISE and the component that provides the service interfaces to the trusted systems. At a high level, the ISI features are captured in three functional areas: Entitlement Management, Information Management, and Logging/Auditing.

The features of the functional areas are:

Entitlement Management

ENT-001. Individual users are granted access to maritime information based on information access policies and attributes assigned by the information providers.

ENT-002. The ISI recognizes a standardized set of security attributes for entitlement management as defined in the National Architecture Plan. Information providers use the security attributes to tag information they publish to the ISI with metadata to convey data protection requirements. The security attributes are included as metadata on the information provided to consumer systems for integrity throughout the data management lifecycle.

ENT-003. Information providers can apply security attributes at varying levels of granularity depending on policy: message-level (e.g., an entire XML message); record-level (e.g., records within an XML message).

ENT-004. Trusted systems authenticate users using either an internal or external identity management process. Trusted systems supply user attributes for all queries. The user attributes and entity attributes (corresponding to the trusted system) are compared to the information provider supplied security attributes on the data to make entitlement decisions.

ENT-005. Users are prohibited from accessing information for which they or their trusted system do not have entitlements.

ENT-006. Both information provider and information consumer trusted systems are authenticated by the ISI. All connections between the ISI and the trusted systems will be secured via SSL

ENT-007. Trusted systems will be responsible for complying with all rules for maintaining trusted system status.

Information Management

INF-001. Ingest, index, and cache messages from all information providers.

INF-002. Information providers can set the expiration date of their messages, which shall not exceed 30 days from the time of publication; the ISI will automatically remove expired messages from the cache when they expire.

INF-003. Search and retrieve information from the cache to service requests from trusted systems.

INF-004. Securely dispose of expired information.

Logger & Auditing

LOG-001. Log all external interactions with the ISI.

LOG-002. Log all internal system activities.

LOG-003. Log all attributes presented by a trusted system for each request, the request itself, the messages retrieved to match the request, and the resulting messages returned to the trusted system upon an entitlement decision.

LOG-004. Log all attempted and established connections and sessions.

LOG-005. Capture all states and activities necessary for auditing.

LOG-006. Log all message disposal activity.

LOG-007. Maintain all logs in accordance with all auditing requirements. Ensure that logs are fully backed up for auditing purposes.

LOG-008. Log all auditing activity.

5. Maritime Information Sharing Environment

The processes required for the GMCOI to both share and protect information are explained in this section. There are key components of the MISE that are critical to understanding before explaining how a member of the GMCOI becomes a participant or trusted system. The figure below is a physical view of the systems and roles in the MISE.

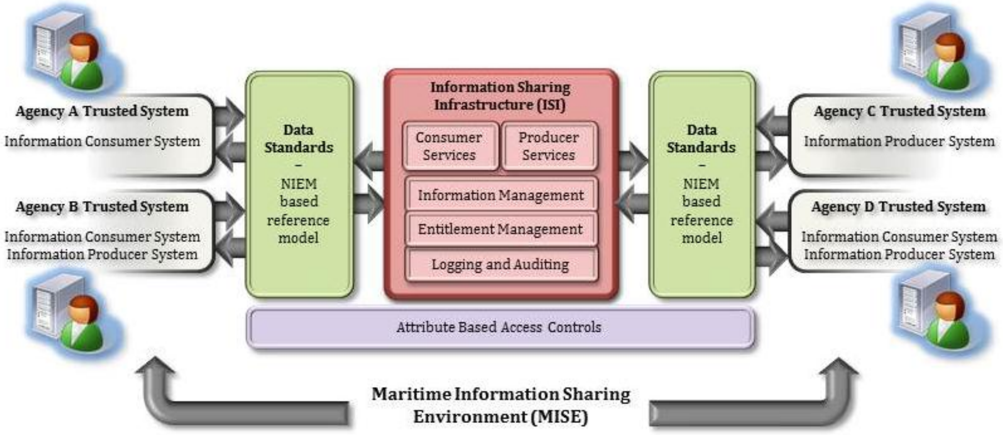

Agency A Trusted System
Information Consumer System
Agency B Trusted System
Information Consumer System
Information Producer System
Data Standards
-
NIEM based reference model
Information Sharing Infrastructure (ISI)
Consumer Services
Producer Services
Information Management
Entitlement Management
Logging and Auditing
Data Standards
-
NIEM based reference model
Agency C Trusted System
Information Producer System
Agency D Trusted System
Information Consumer System
Information Producer System
Attribute Based Access Controls
Maritime Information Sharing Environment (MISE)

Figure 16: MISE Physical View

The *MISE Physical View* illustrates several key characteristics of the MISE, and introduces some terminology:

- The MISE consists of many separate systems which are owned and operated by separate agencies. These systems and agencies work together in a *federation*. In most cases these are existing systems, already fulfilling key roles within their respective agencies. To participate in the MISE, they will be expected to adhere to MISE specifications, allowing them to share information within or retrieve information from the information sharing environment.
- The *Information Sharing Infrastructure (ISI)* is a system operated by MISE Management, and acts as the central hub of the environment. The ISI provides essential services to the systems within the MISE. The role of *MISE Management* is discussed in the appendices.
- Each system within the MISE is referred to as a *Trusted System*.
- To participate in the MISE, trusted systems connect to and interact with the ISI, not directly with other trusted systems. Network connections between trusted systems and the ISI are over the public Internet.
- *Information Provider* and *Information Consumer* are the two key roles which a trusted system can take within the MISE. Any trusted system may serve as either or both of these roles, as illustrated in the figure *MISE Physical View*.

- An *Information Provider System* makes information available to other trusted systems, and maintains control over which permissions users must possess to access the information. The information provider system controls access by providing security attributes to the ISI for each record published.
- An *Information Consumer System* retrieves information from the MISE. The consumer system is given access to information within the MISE based upon the applicable *Information Access Policy*.

Users in the MISE are associated with a specific trusted system which conducts interactions such as authenticating the user and retrieving information from the ISI on behalf of the user. This trusted system is the conduit for the user to access information from the ISI or other systems in the MISE.

6. National Maritime Architecture Framework

Maritime information must be integrated and shared across numerous stakeholder agencies to gain an effective understanding of the maritime domain. To allow sharing and integration across organizational boundaries, information providers must be assured that their information is protected with access granted only to those parties entitled to the information in accordance with applicable laws and regulations. The National Maritime Architecture Framework is an interoperable solution and approach to enable stakeholders across the GMCOI to share and protect their information.

The National Maritime Architecture Framework is comprised of four architectural views:

1. Data Architecture View
2. Services Architecture View
3. Security Architecture View
4. Technical Operations Architecture View

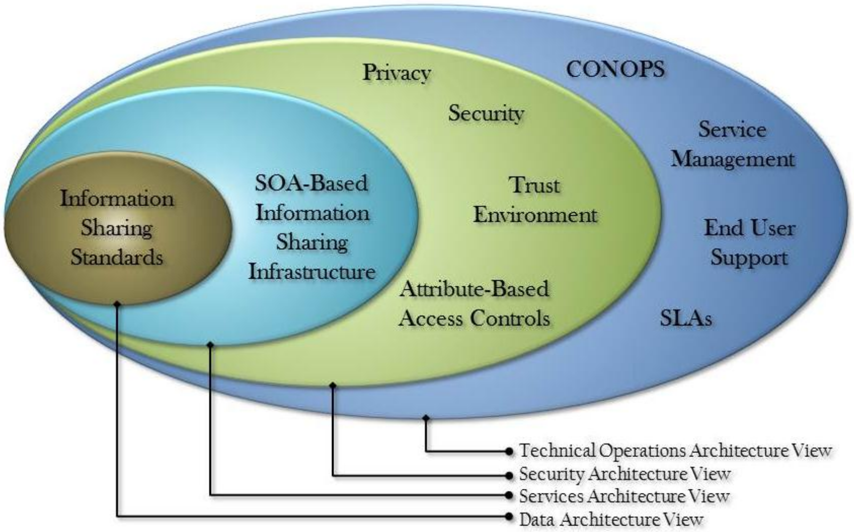

Privacy
CONOPS
Security
Service Management
Information Sharing Standards
SOA-Based Information Sharing Infrastructure
Trust Environment
End User Support
Attribute-Based Access Controls
SLAs
Technical Operations Architecture View
Security Architecture View
Services Architecture View
Data Architecture View

Figure 17: The Four Architectural Views of the Framework

Data Architecture View

Shared vocabulary and information sharing standards based on the National Information Exchange Model – Maritime (NIEM-M) reference models form the foundation of the framework. In accordance with the National MDA Architecture Hub Strategic Plan, the Maritime Framework leverages NIEM for a common vocabulary and consistent processes to build maritime information exchange standards.

Services Architecture View

The information sharing infrastructure (ISI) is the core functionality for the Maritime information sharing environment. The ISI provides a common integration platform to facilitate management and dissemination of maritime information across the maritime community. The categories of services within this platform that are provided by the ISI include: Information Brokering, Request Brokering, Logging, Auditing, Information Management, and Entitlements Management.

Security Architecture View

The Trust Environment employs standards-based Federated Identity Management for secure, seamless access to maritime information products, which are then shared among trusted systems and the ISI. The Trust Environment consists of attribute-based access control, common user attributes/entitlements metadata, SAML[6] profile for entitlement assertions, and fine-grained, information provider managed access control policy. This environment ensures only trusted systems interact with the ISI and that user entitlements are determined based on user attributes.

Technical Operations Architecture View

The Technical Operations View defines how the framework as a hole is managed and operated. It includes governance and agreements used to manage the MISE.

6.1. DATA ARCHITECTURE VIEW

6.1.1. INFORMATION MODEL

While initial focus is on the four national information products[7], a guiding design principle allows other types of information to be handled without requiring modification to participating trusted systems.

The *information model* consists of a set of characteristics which are true of all shared information. These characteristics are:

1. A *record* is the unit of shared information. Records are represented as XML documents.

[6] Security Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between systems

[7] Advance Notice of Arrival (NOA), Indicators and Notifications (IAN), Positions and Tracks, and Maritime Operational Threat Response (MOTR)

4. Each record has a *record type*, which is defined by a NIEM IEPD (discussed below). The record type can be equated to a record subject; Vessel, People, Notification, etc.

5. A *Data Set* is a set of records shared into the environment by an information provider. All records within a data set have the same record type. It is possible for an information provider to share more than one data set, if they have multiple types of records to share. Conversely, multiple information providers may share data sets of the same record type.

6. An *Information Access Policy* defines the access to records within the data set. Information providers convey information access policy by applying security attributes at the record logical block, or element level in the records they publish.

The ISI and information consumer systems define and build services based on knowledge of this information model and record type definitions, while relying on the consistent representation of information provided by the NIEM-M vocabulary. Specialized functionality can and will be built by information consumer systems based on detailed understanding of specific record types. The information model concept is unique because it allows a base level of functionality to operate upon record types defined and introduced after the information consumer system is implemented. (The concept is similar to polymorphism in object oriented programming, i.e. code written to operate on objects of a base type can still deliver some useful functionality on objects of a more specialized derived type.)

6.1.2. NIEM

The National Information Exchange Model (NIEM) is an XML-based information exchange framework for sharing data between United States government agencies and their information partners.

The NIEM facilitates the creation of automated enterprise-wide information exchanges which can be uniformly developed, centrally maintained, quickly identified and discovered, and efficiently reused. The result aims toward more efficient and expansive information sharing between agencies and jurisdictions; more cost-effective development and deployment of information systems; improved operations; better quality decision making as a result of more timely, accurate, and complete standardize information; and, as a consequence, enhanced public safety and homeland security.

The NIEM framework has several components:

- NIEM Core: A common XML-based data model that provides data components for describing universal objects such as people, locations, activities, and organizations.
- Domains: More specialized XML data models for individual use cases. There is a specialized domain for Maritime, along with a number of others (e.g. Justice and Immigration).
- Information Exchange Package: A methodology for using and extending the building blocks that comes from the common and domain-specific models, and turning them into a complete information exchange.
- Tools: Help develop, validate, document, and share the information exchange packages.
- Governance Organization: Provides training and support and oversees NIEM's evolution over time.

The NIEM-M XML vocabulary provides a combination of objects from NIEM core, the Maritime domain, and additions via the EIEM and IEPD, described in the next sections.

6.2. ENTERPRISE INFORMATION EXCHANGE MODEL

An Enterprise Information Exchange Model (EIEM) is a NIEM-conformant set of XML schemas and other artifacts. The EIEM defines core entities, also termed Business Information Exchange Components (BIECs) that serve as building blocks that are reused in many maritime exchanges.

The high-level BIECs defined by the Maritime EIEM are:

- CDC Cargo (Certain dangerous cargo as declared in an Advance Notice of Arrival)
- Crew Nationality Count
- GeoLocation
- Interest
- NOA Cargo (Bulk cargo as declared in an Advance Notice of Arrival)
- Non Crew Nationality Count
- Port Visits
- Position
- Record Metadata
- Vessel Information
- Vessel Characteristics

Using an EIEM increases the commonalities among models, which has a number of benefits:

1. Simplifies development and allows users to develop reusable processes that work across multiple record types.
2. Reduces complexity by reducing the number of overall elements that need to be supported.
3. Decreases the time and cost of maintaining the model because it is simpler and has less overlap.

6.3. INFORMATION EXCHANGE PACKAGE DOCUMENTATION

An Information Exchange Package Documentation (IEPD) defines a particular record type, the basic unit of shared information. The first IEPDs for the initial focus areas are:

- Advance Notice of Arrival (NOA)
- Indicators and Notifications (IAN)
- Vessel Positions and Tracks
- Maritime Operational Threat Response (MOTR)

As depicted in the figure below, an IEPD uses a subset of the core BIECs defined in the EIEM and assembles them into a particular record type with its own unique root element. As new requirements are defined, new IEPDs can be created that build on the same EIEM core entities.

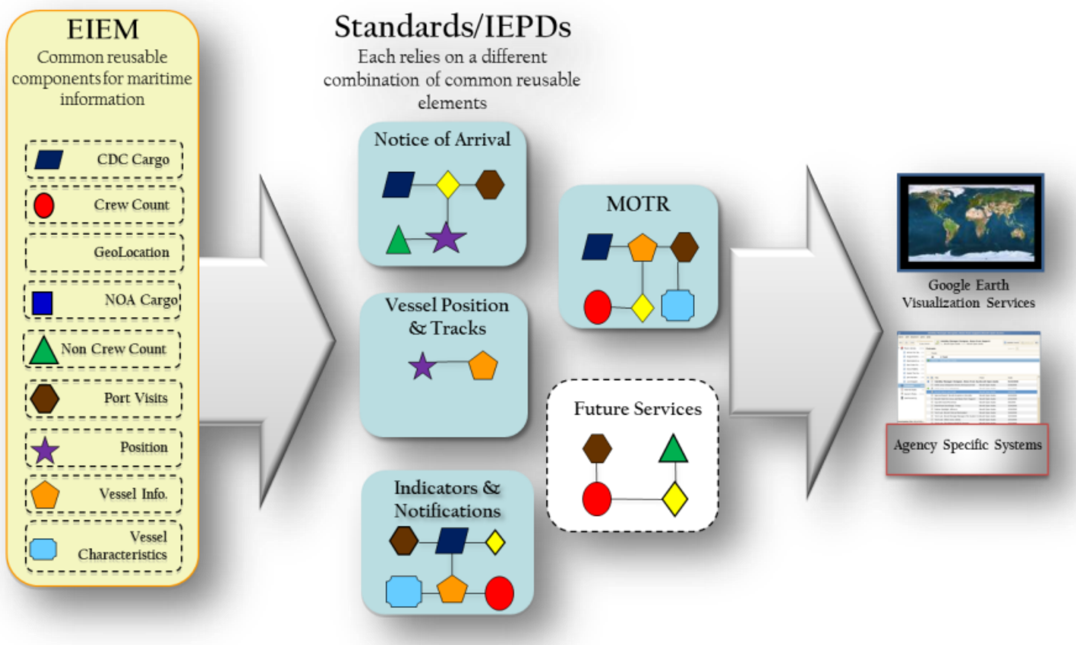


Figure 18: IEPDs combine core entities from the EIEM

The MISE is flexible enough to allow an IEPD to extend or restrict the EIEM components, as long as no required properties are removed. An IEPD can also define new exchange-specific information, for example a notice date for a notice of arrival.

Physically, an IEPD is a collection of artifacts that describe the exchanged data, including:

- XML Schemas that describe the structure of the XML documents:
    - *Exchange schema* that declares the root element of the exchange
    - *Extension schema* that contains exchange-specific components
    - Copy of the EIEM schema that defines the BIECs
- Documentation artifacts such as a schema description and change log
- Metadata artifacts such as rendering instructions, and information to aid query building
- Sample XML instances

6.4. SERVICES ARCHITECTURE VIEW

The Architecture Plan defines the MISE service interfaces which allow trusted systems to retrieve and publish information. Trusted systems interact with the ISI using these service interfaces.

6.4.1. SERVICE TYPES

RETRIEVING DATA

Two services are defined for retrieving information:

- **Search Service**: Find records matching specified criteria. Returns a list of summary records and record IDs in an atom feed.
- **Retrieval Service**: Retrieve a record with a specified ID. Returns the NIEM-M representation of the record.

These services provide access to records, which, are the unit of information the MISE operates upon. Information consumer systems may choose to present a relatively low level user interface allowing search, retrieval, and viewing of individual records. Information consumer systems may also use information obtained from multiple service invocations to compose a higher-level operational picture for presentation to users.

PUBLISHING DATA

In addition, a single service is defined for *publishing* information into the ISI.

- **Publication Service**: Place information into a cache, and keep the cache up to date including delete operations to remove information from the cache.

Each of these services can be thought of as having a *provider* side and a *consumer* side. Providers *implement* the service and make information available. Consumers *use* the service to obtain information.

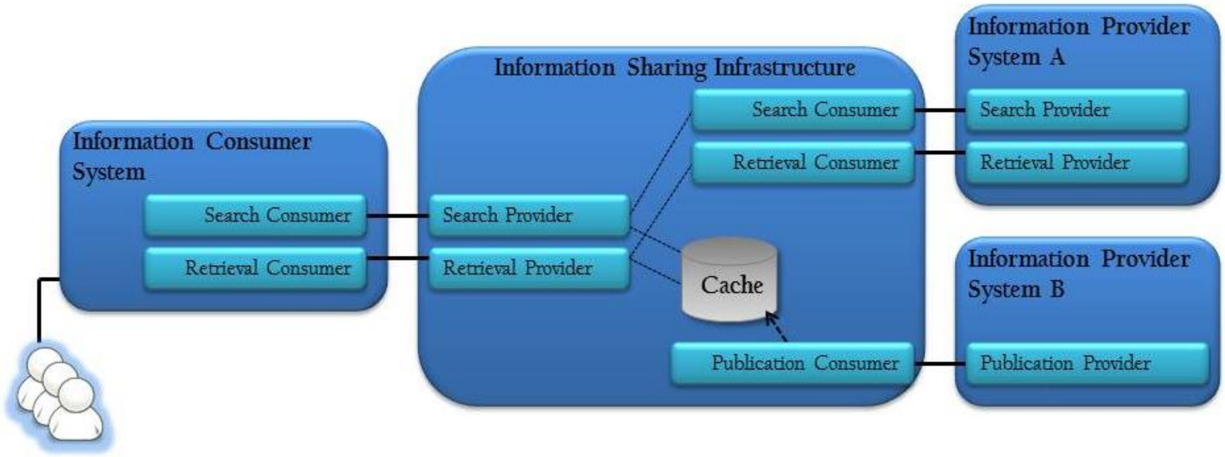


Figure 19: Service Providers and Consumers

The figure above illustrates several important aspects of MISE services:

1. From a services perspective, information providers have two integration choices.

   a) ***Information* Broker**: Information Provider B (figure above, bottom right) has chosen to publish information to the ISI cache, so it implements only the provider side of the Publication service interface. The ISI is able to provide Search and Retrieval services to information consumers using the cached information. In this case, the ISI acts as an *information broker*.

b) ***Request*** **Broker**: Information Provider A (figure above, top right) has chosen not to cache information at the ISI. In this case, each time an information consumer system searches the ISI using the Search service, the ISI must in turn search the information provider system. The same is true for Retrieval. So in this case, the information provider system must implement the provider side of the Search and Retrieval services, and the ISI implements *federated search* and *brokered retrieval*. In this case, the ISI acts as a *request broker*.

2. The same service interfaces are used in different contexts. For example, information provider systems which do not use the ISI cache provide the same Search interface to the ISI as the ISI provides to information consumer systems. Depending on the role of a trusted system (information provider and/or consumer), and its integration choices, provider and/or consumer sides of several MISE service interfaces may need to be implemented.

The ISI serves as the common integration point, allowing each trusted system to make integration choices, while still enabling interoperability.

6.4.2. SERVICE CHARACTERISTICS

This section describes characteristics which are common across all service interactions between trusted systems and the ISI. Subsequent sections provide more details of each service.

NETWORK CONNECTIVITY

Trusted systems connect to the ISI over the public Internet. All service interactions use the HTTPS protocol.

RESTFUL

MISE services are RESTful web services[8]. RESTful web services use the HTTP protocol as an application protocol, rather than simply as a transport. RESTful web services are implemented very much like standard web applications, as opposed to using any type of complex middleware layer. The RESTful style was chosen for MISE because of its simplicity and ease of integration.

As RESTful services, nearly all MISE service interactions follow a simple pattern, which is the same pattern used by a web browser accessing a page on the Internet:

- The service consumer requests an HTTP GET of a URL specifying a resource at the service provider. The URL is known to the service consumer because it is one of the following:
    1. A top-level URL of the service provider obtained from configuration information.
    2. Relative to the top-level URL, as specified in the MISE Search/Retrieve interface specification.
    3. Obtained from the body of an earlier response from the service provider.

[8] The term “REST” stands for REpresentational State Transfer and is an is an architecture style for designing networked applications

- The service provider returns a resource in response to the GET. The format of the resource is specified in the MISE Service/Retrieve Interface Specification. In most cases it is an XML document adhering to a specified schema or format.
- Any error conditions are reported using standard HTTP status codes.
- Other aspects of the exchange (e.g. session handling, conditional get, content negotiation) are handled following standard HTTP conventions.

AUTHENTICATION AND ENTITLEMENT

All MISE services operate in the context of the security architecture discussed in section 4.5. For every service interaction, provider and consumer systems are authenticated. If a service invocation occurs on behalf of a user, authenticated attributes of the user are available. From the standpoint of service definitions, the security architecture operates conceptually when the connection is established, not as a part of each HTTP request/response message.

6.4.3. SEARCH SERVICE

The Search service allows a consumer to locate records which match specified criteria. The Search service provides a flexible query language. In most cases, the search service does not return full records, but instead returns the IDs of matching records. If a consumer wishes to obtain a full record, an ID obtained from the search service is used with the Retrieval service to obtain it.

The core of the Search service interface consists of:

- The Search consumer requests an HTTP GET of a URL such as https://*providerbase*/search?q=*query*, where *providerbase* is a base URL obtained from configuration information, and *query* provides search criteria expressed using the Search service query language, which is specified in detail in the MISE Search/Retrieve Interface Specification . The query language allows searching for records containing elements from the NIEM-M vocabulary with specified values. It also allows constraining results to specific record types, providers, and data sets.
- The Search provider responds with an Atom document (which is an XML document following a specific schema), listing IDs (which take the form of URLs) of matching records, and other information described in the MISE Search/Retrieve Interface Specification. To be included in the list, the portion of the record which the user is entitled to see (according to the IAP) must match the search criteria. If a large number of records match the search criteria, only a subset is included to constrain the size of the response. In this case, the response also includes the total match count, and a URL to the next page of results.

Information consumer systems often have a need to monitor the MISE for new information matching specified criteria. This capability is provided using the *web syndication* model. The information consumer system periodically queries the Search service, using the full search query language, but including additional parameters allowing the returned Atom document to report recent changes in a manner efficient for the provider and consumer.

The ISI also supports some specially-defined *value-added* searches, allowing related information from a number of records to be combined, and returning full information in response to the search request. In this case, it is not necessary to use the Retrieval service.

Another part of the Search service interface specifies relative URLs to metadata describing data sets, information providers, and record types which are available at the Search provider. This information is used by Search consumers to form proper queries, and to allow building of a search user interface.

6.4.4. RETRIEVAL SERVICE

The Retrieval service obtains the full NIEM-M representation of a specified record. The Retrieval service interface consists of:

- Retrieval consumer requests an HTTP GET of a URL representing a specific record. This URL will have been obtained using some previous service invocation, for example from Search results.
- Retrieval provider responds with the NIEM-M representation of the specified record. The Retrieval provider ensures that the IAP is applied so consumers are provided information at their level of access.

An information consumer system may use the Retrieval service for the purpose of building a higher-level operational picture, in which case it may not be necessary to display the individual record to a user. In other cases, an information consumer system may display the record in a manner appropriate and intuitive for individual viewing. Metadata associated with the record type, containing instructions for converting the NIEM-M representation to HTML, can be read by information consumer systems.

6.4.5. PUBLICATION SERVICE

The Publication service allows an information provider to maintain an up-to-date copy of shared information at another location within the MISE. The Publication service interface is used by information providers to update the ISI cache, enabling the ISI to provide Search and Retrieval services on their behalf. In this case, the ISI is the Publication *consumer*, and the information provider is the Publication *provider*.

A Publication provider makes three categories of resources available, which can be accessed by a Publication consumer using HTTP GET:

1. **Metadata** – lists available data sets and record types.
2. **List of all shared records** – resource(s) in Atom format listing IDs of all shared records in the data set.
3. **Ordered list of changes** – resource(s) in Atom format listing all changes to the data set (new records, deleted records, changed records).

These categories of resources provide all information necessary to allow the Publication consumer to initially obtain a full copy of the shared data set, and also to efficiently check for changes. The Publication consumer reads these resources using HTTP GET.

6.4.6. HOW THE SERVICES WORK TOGETHER

The following example serves as an example of how the MISE services can be used by information consumer systems to deliver value and insight to users and information consumer system displays graphic showing incoming vessels within a region surrounding a port. Vessels which are mentioned in IAN or MOTR records are overlaid with an icon to make this visually apparent. Users of the system can click on a vessel and drill down to see detailed information about the vessel. Users can also choose to display tracks showing the path taken by the vessel to reach its current position.

To implement this functionality, the information consumer system would interact with the MISE in the following manner to obtain information needed:

- A key aspect of implementing this functionality is creating and maintaining the collection of information needed to generate the graphic. This information would be initially obtained using the Search service with a query requesting all records of type NOA, IAN, MOTR, or POS, with location inside the area of interest (region surrounding specified port). The system would then use the Retrieval service to obtain full records for all matches listed in the search response.
- In order to keep the graphic up-to-date, the system must keep its internal representation of the above information up-to-date. To do so, it periodically uses the Search service with the same query as above, but with additional parameters allowing efficient detection and reporting of changes in the returned Atom document.
- Since the information consumer system wishes to display this graphic to many users, without the overhead of separately searching on behalf of each user, service requests are associated with a set of user attributes representing a group of users. The information consumer system is responsible for only displaying the graphic to individual users who possess these same attributes. If the system wishes to also display a version of the graphic generated using more privileged information to more privileged users, a separate Search can be requested with a more privileged set of user attributes to maintain this separate internal representation.
- If a user chooses to display the track showing the path taken by a vessel, the system will have the updated position information available in its internal representation.
- If a user clicks on a vessel to drill down and see all available information, the system can show information from the internal representation (e.g. NOA) immediately. The system should also at that point make a Search request for all records of *any* type mentioning the vessel. In this manner, as the MISE expands to include additional types of information (record types), the information consumer system will be able to display this information to authorized users *without the need to modify each information consumer system when new record types are introduced.*
- In a similar manner, if a MOTR record exists and the vessel has been boarded, the system could use the Search service to find any biometric information which has been gathered and the user is authorized to view.
- If an information consumer system was developed with a built-in knowledge of specific IEPDs for known record types, it can display information from those records in any manner it sees fit. To display a record for a user *without* having a built-in knowledge of the record

type, the information consumer system can read the metadata associated with the record type from the ISI to obtain instructions allowing generation of HTML presenting the record contents in a manner meaningful to a human viewer.

6.5. SECURITY ARCHITECTURE VIEW

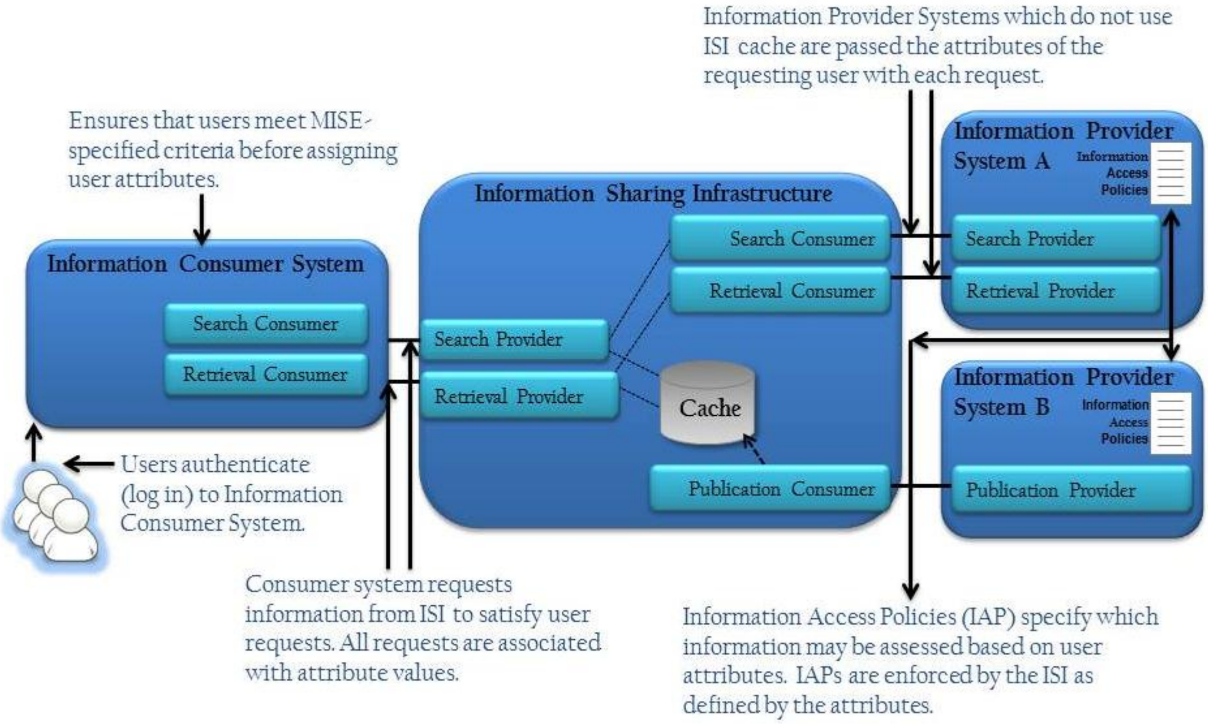
The purpose of the MISE Security Architecture is to ensure that shared information is protected in accordance with applicable doctrine. Information providers must be assured that shared records will only be revealed to users who meet certain criteria. In addition, some records contain subsets of information which should only be revealed to a subset of MISE users. For example, certain portions of records containing privacy protected information must only be revealed to users who have been authorized to view such information.

This section describes the access control system which meets these requirements.

6.5.1. FUNCTIONAL PERSPECTIVE

We begin by describing the MISE *attribute-based access control system* from a logical or functional perspective. The *Trust Implementation Perspective* section below focuses on how it is secured.

The figure below illustrates key functional aspects of MISE access control.



Information Provider Systems which do not use ISI cache are passed the attributes of the requesting user with each request.

Ensures that users meet MISE-specified criteria before assigning user attributes.

Users authenticate (log in) to Information Consumer System.

Consumer system requests information from ISI to satisfy user requests. All requests are associated with attribute values.

Information Access Policies (IAP) specify which information may be assessed based on user attributes. IAPs are enforced by the ISI as defined by the attributes.

Figure 20: Security Architecture - Functional Perspective

USER ATTRIBUTES

As discussed in section on ***Service Types***, information provider systems share data sets with the MISE, and information consumer systems provide users access to the shared information. Users authenticate (log in) to their information consumer system, which then obtains records from the

ISI on behalf of the user using the services discussed in section on ***Service Types***. Users authenticated by many different consumer systems access information shared by many different provider systems. Providers must have a way to control which users can access which portions of shared information, but it is not practical for providers to possess knowledge of individual users, or even of individual information consumer systems. Rather, a standardized way of describing users is necessary. This is the purpose of standardized *User Attributes*.

The MISE defines a common set of user attributes. Information providers convey *Information Access Policies* by applying security attributes to the records they publish specifying access is allowed for users who possess certain of these attributes, rather than to specific users or to specific information consumer systems. When obtaining information on behalf of a user, information consumer systems associate the set of attributes possessed by the user with the request. Information consumer systems are responsible for user vetting – i.e., ensuring that the attributes for a user are accurate and meet the precise meaning of the attribute as defined by the MISE. Information consumer systems are also responsible for authenticating users, usually by requiring entry of a password or by requiring possession of a token or key.

Specifically, the MISE defines the following user attributes for use by information access policies, referred to as *entitlement user attributes*:

- **Citizenship Code** – The country that has assigned rights, duties, and privileges to the user because of the birth or naturalization of the user in that country.
- **LawEnforcement Indicator** – User required and qualifies for law enforcement information.
- **PrivacyProtectedIndicator** – User requires and qualifies for privacy protected information.
- **COIIndicatior** – Minimum access level assigned; user requires and qualifies for information shared by MISE community.

Other user attributes are defined for auditing purposes, referred to as *auditing user attributes*. They are not used by information access policies, but are recorded in audit logs as a record of the user who accessed information. The auditing user attributes listed below are mandatory. Additional optional attributes are defined in the MISE Attributes Specification.

- **FullName** – The complete name of the user.
- **ElectronicIdentityId** – The unique identifier that is associated with the user within the user's information consumer system.

INFORMATION ACCESS POLICIES

Each shared data set has an associated *Information Access Policy (IAP)*. The IAP is specific to each data set and is maintained by the information provider. An IAP is a rule set that define attributes users must possess to access each logical block within records in the data set. IAPs allow denying access to entire records, or to portions of records. Rules may be conditional based on values within the record. Note that IAPs apply to all information consumer services (Search and Retrieval). During Retrieval, access to entire records may be denied, or unauthorized portions of records may be excluded from the returned NIEM-M document. Search behaves exactly as if only authorized records or portions of records existed at the provider. The process of enforcing the IAP as user's access information is referred to as the *entitlement decision*.

As previously discussed in the ***Services Architecture View*** section, information providers may choose to publish to the ISI cache, allowing the ISI to implement information consumer services on their behalf, or to implement those services themselves. This distinction also applies to the enforcement of IAPs. Information providers which do not publish to the ISI cache must enforce information access policies. This concept is covered in detail in the National MDA Architecture Feature Spec document. Whether enforced by the ISI and/or the information provider, user attributes that are necessary for making the entitlement decision are made available as discussed in above ***User Attributes*** section.

ENTITY ATTRIBUTES

Just as user attributes describe characteristics of users in a standardized way, *entity attributes* describe characteristics of trusted systems within the MISE. (*Entity* is SAML terminology for a *trusted system*. The usage of SAML in MISE is covered in the section ***MISE Certificate Authority***). And just as a user's information consumer system is responsible for user vetting, MISE Management Organization is responsible for trusted system vetting – i.e., confirming that the trusted system has met all criteria necessary for joining the MISE and that all entity attributes associated with the trusted system are accurate.

The MISE defines the following entity attributes which are involved in entitlement decisions, referred to as *entitlement entity attributes*:

- **EntityId** – The unique identifier by which the entity (system) is known.
- **LawEnforcementIndicator** – Entity (system) authorized to handle law enforcement information.
- **PrivacyProtectedIndicator** – Entity (system) authorized to handle privacy protected information.
- **COIIndicator** – Entity (system) authorized to handle information shared by MISE community.

The *Indicators* associated with an entity is the full set of *Indicators* the entity is authorized to assign to users. This permits the limiting of the authority of a trusted system. For example, if an information consumer system requests information on behalf of a user and presents an *PrivacyProtectedIndicator* user attribute, but the information consumer system itself *does not* have an *PrivacyProtectedIndicator* entity attribute, the request will be denied by the Information Sharing Infrastructure.

The following entity attributes are examples of those defined for auditing purposes, referred to as *auditing entity attributes*. They are not relevant to entitlement decisions, but are recorded in audit logs as a record of the trusted system which accessed information:

- **EntityName** – The name of the entity in a format suitable for display to a user.
- **OwnerAgencyName** – The name of the organization or agency by which the entity is owned.

- **OwnerAgencyCountryCode** –The country of the organization or agency by which the entity is owned and operated.

Additional entity attributes are defined providing technical and administrative point of contact information for the trusted system. Full details can be found in the MISE Attributes Specification.

AUDIT LOGGING

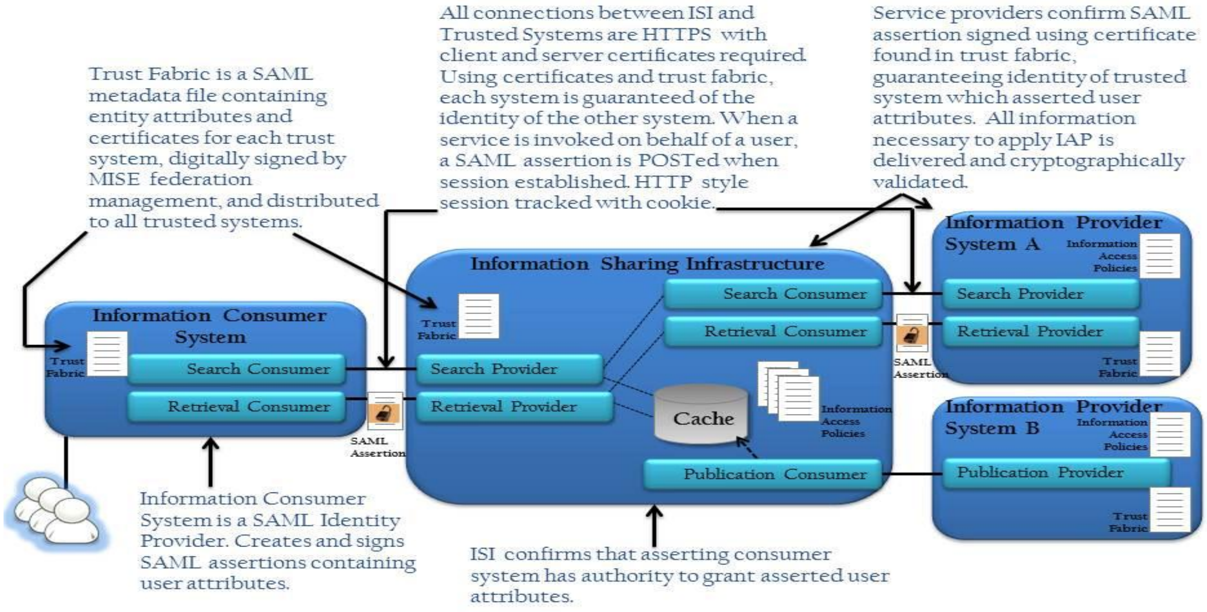
The ISI maintains audit logs that record all accesses to information from all information consumer systems to include: details regarding the request, specific information provided, criteria used to grant access, and *auditing attributes* tracking the trusted system and (when relevant) the user which requested the information.

Information consumer systems may request information on behalf of individual users, as well as on behalf of a group of users who all have an identical set of entitlement user attributes. In the former case, auditing entity attributes and auditing user attributes are recorded in audit logs. In the latter case, auditing user attributes are not recorded, and the information consumer system is responsible for revealing the information only to users with the precise set of auditing user attributes.

6.5.2. TRUST IMPLEMENTATION PERSPECTIVE

The *Functional Perspective* section focuses on MISE's *attribute-based access control* – on how user attributes, entity attributes, and information access policies are used to control which information is accessible by which users, and to audit actual accesses. This section, focuses on the cryptographic methods used to establish trust and deliver attributes in a secure manner for making entitlement decisions.

The figure below illustrates key aspects of the trust implementation.



Trust Fabric is a SAML metadata file containing entity attributes and certificates for each trust system, digitally signed by MISE federation management, and distributed to all trusted systems.

All connections between ISI and Trusted Systems are HTTPS with client and server certificates required. Using certificates and trust fabric, each system is guaranteed of the identity of the other system. When a service is invoked on behalf of a user, a SAML assertion is POSTed when session established. HTTP style session tracked with cookie.

Service providers confirm SAML assertion signed using certificate found in trust fabric, guaranteeing identity of trusted system which asserted user attributes. All information necessary to apply IAP is delivered and cryptographically validated.

Information Consumer System is a SAML Identity Provider. Creates and signs SAML assertions containing user attributes.

ISI confirms that asserting consumer system has authority to grant asserted user attributes.

Figure 21: Security Architecture - Trust Implementation Perspective

MISE CERTIFICATE AUTHORITY

MISE Management will use a certificate authority (CA) to sign the trust fabric. Certificates signed by this certificate authority form the cryptographic foundation of trust within the MISE.

Each trusted system uses a signed certificate and matching private key to secure SSL connections. In addition, each information consumer system uses a separate signed certificate and matching private key to sign SAML assertions. The following sections provide more details on the usage of these certificates.

TRUST FABRIC

The MISE *Trust Fabric* is a digitally signed XML document in SAML metadata format describing each trusted system in the MISE. This document is signed by the MISE Certificate Authority, and distributed to all trusted systems as a trust anchor upon which all cryptographic operations rely.

In SAML metadata terms, the trust fabric document contains an EntityDescriptor element describing each trusted system, and also an EntityDescriptor element describing the ISI itself. Two types of RoleDescriptor elements are used:

1. **Trusted System** – each EntityDescriptor contains a RoleDescriptor of this type. Essential associated information includes the certificate used for SSL connections, as well as *auditing entity attributes*.
2. **Identity Provider** – information consumer systems have an identity provider role *in addition to* the trusted system role. Essential associated information includes the certificate used for signing SAML assertions, and the appropriate *InformationAccessRights* entity attributes introduced above.

The organization owning each trusted system works with MISE Management to prepare the EntityDescriptor information describing the trusted system. When EntityDescriptor information is complete, has been validated for accuracy, and confirmed to meet all MISE criteria, MISE Management incorporates it into the trust fabric document, signs the document with the MISE CA private key, and distributes the updated trust fabric to all trusted systems. The same process is used when any aspect of the EntityDescriptor information for any trusted system changes.

The trust fabric document is made available for retrieval at any time by trusted systems from a well-known URL on the ISI, HTTPS is required to GET the trust fabric document, , but a client certificate is no required. All trusted systems must retrieve the trust fabric on a periodic basis, to ensure they are always operating with the current version of the trust fabric. MISE is able to use this simple distribution mechanism for the trust fabric since the security of the MISE does not rely on the contents of the trust fabric document being kept secret, but rather upon its accuracy being guaranteed by the CA signature.

NETWORK CONNECTIONS

All network connections between the ISI and trusted systems are over the public Internet, using HTTP over SSL (HTTPS). Client and server certificates, digitally signed by the MISE certificate authority, are required for each SSL connection.

When a network connection is established, the ISI and trusted system each use the certificate submitted in the trust fabric, and make the associated private key available to SSL software. For the purpose of SSL connections for MISE service interactions, all trusted systems must install the MISE CA cert as a trusted root CA. After the connection is established, each side can then obtain the certificate used by the other side from SSL software, and confirm it is found in the trust fabric. If it is not, the connection should be immediately dropped. Possession of the private key associated with a certificate signed by the MISE CA is thus proven, and lookup in the trust fabric proves the identity of the trusted system and that it is a current validated member of the MISE. Note that among its other roles, the trust fabric serves the purpose of the certificate revocation list which is a part of some PKI arrangements.

This type of network connection and certificate validation allows certain security and authentication issues to be addressed at connection time, allowing simple RESTful style application level message exchanges. Specifically, the following are addressed:

- **Trusted System Authentication** – For every message exchange between the ISI and a trusted system, each system will verify the identity of the other system.
- **Message Non-repudiation and Integrity** – Every message received by the ISI or a trusted system will be verified to prove the Trusted System that claimed to have sent the message actually sent it and the message was not altered since it left control of the Trusted System that sent it, otherwise the message will be rejected.
- **Message Confidentiality** – Every message sent or received by the ISI will be cryptographically protected from being read by any person or entity other than its intended receiving system.

USER ATTRIBUTE CONVEYANCE

This section describes the mechanics of delivering authenticated user attributes to the right location at the right time, allowing entitlement decisions to be made.

Information consumer systems associate a SAML assertion with every request for information from the MISE. Information consumer systems are the authoritative source of user attributes, since they are responsible for user vetting and authentication. In SAML terms, information consumer systems act as *identity providers*. This does not preclude information consumer systems from relying on an external identity provider, including acting as a relying party to another SAML identity provider.

Characteristics of this SAML assertion include:

- Contains the appropriate authenticated set of entitlement user attributes and auditing user attributes.
- Digitally signed by the information consumer system using the private key associated with the signing certificate which is a part of its *Identity Provider* role information within the trust fabric document. This provides a cryptographic guarantee to the ISI and to information provider systems of the identity of the information consumer system asserting the user attributes and promising to handle the information in accordance with MISE guidelines.

- Contains a SAML AudienceRestriction specifying the entire MISE, meaning it is intended for interpretation by the ISI and also by information provider systems.

The figure below illustrates the mechanism used for associating this SAML assertion with a series of RESTful HTTP request / response messages making up service interactions.

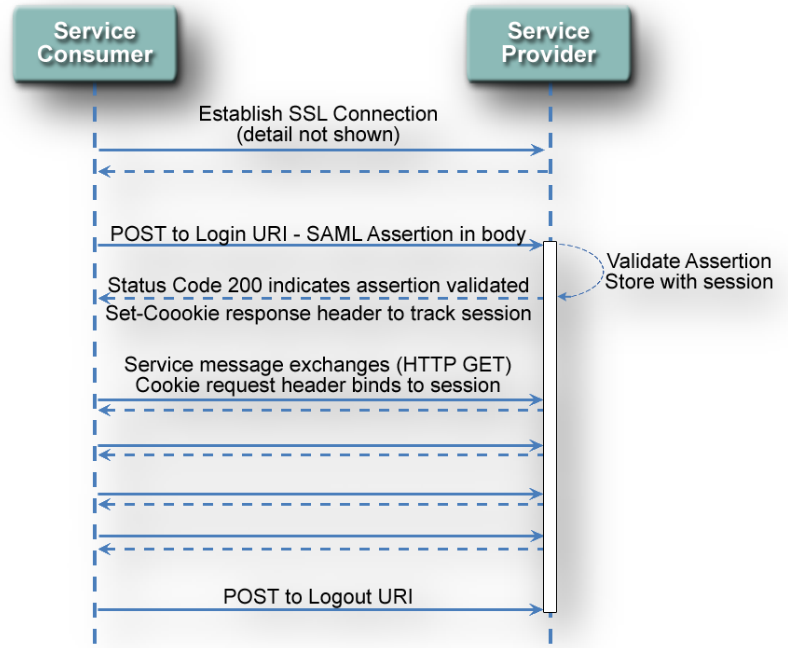

Service Consumer
Service Provider
Establish SSL Connection
(detail not shown)
POST to Login URI - SAML Assertion in body
Validate Assertion
Store with session
Status Code 200 indicates assertion validated
Set-Coookie response header to track session
Service message exchanges (HTTP GET)
Cookie request header binds to session
POST to Logout URI

Figure 22 : SAML assertion bound to HTTP session

Notice the following from the figure *SAML assertion bound to HTTP session*:

- Specifics mentioned in this section occur inside an SSL connection with both systems authenticated as discussed in ***MISE Certificate Authority*** section.
- Before beginning the series of messages making up service interactions (generally a series of HTTP GETs), the service consumer does an HTTP POST to a login URI at the service provider. The body of this POST request contains the signed SAML assertion. During processing of the POST, the service provider validates the assertion. This includes standard SAML validation as defined in the MISE Interface Security Specification, as well as confirming that the signing certificate exists within the current trust fabric. If the service provider is the ISI, another validation step is ensuring that no *Indicators* are assigned which the information consumer system is not authorized to assign.
- If the assertion is valid, the service provider stores the assertion in a session, returns a Set-Cookie response header to track the session, and returns an HTTP status code of 200 (OK). If the assertion is invalid, the service provider returns an HTTP status code of 403 (Forbidden).
- Once the assertion is validated and a session is created, the service consumer can proceed to interact with the provider in accordance with The MISE Search/Retrieve Interface Specifications. A Cookie request header specifying the session cookie assigned by the

provider must be sent with each HTTP request. No additional authorization overhead is borne during each request/response. Normal web browser style session handling and expiration are used, even though this is a system-to-system interaction.

- If the service consumer knows it is finished interacting in the context of the particular set of user attributes, it can POST to a logout URI as a courtesy to the service provider, allowing the provider to delete the session and free up associated resources. If the service consumer does not explicitly logout, the service provider will delete the session automatically after a period of non-use, or if the time condition within the SAML assertion (NotOnOrAfter) is exceeded.

If the ISI is acting as a request broker, in which case the ISI acts as a service consumer to pass the request to an information provider, the same mechanism is used between the ISI and the information provider. The signed SAML assertion received from the information consumer is simply passed verbatim to the information provider, allowing the provider to identify the information consumer system which originated the request and asserted the user attributes, by locating the certificate used to sign the assertion in the trust fabric.

6.6. TECHNICAL OPERATIONS VIEW

The Technical Operations Architecture View defines the approach for operations and sustainment of services that comprise the MISE. The primary goal of the MISE Technical Operations Architecture View is to provide a high-level overview of MISE operational concepts in the following major areas:

- Service Management
- Organizational Roles and Responsibilities
- End User Support

6.6.1. SERVICE MANAGEMENT

Service management is essential for the MISE to achieve reliable performance and meet stakeholder requirements for dissemination of critical maritime information. Service management includes activities such as monitoring availability of information provider and ISI services, registration of trusted systems for authentication and discovery services, and review of services levels to ensure service performance thresholds are met.

SERVICE LEVEL AGREEMENTS

Service level agreements (SLA) are necessary for each trusted system that connects to the ISI. The service level agreement is a formal agreement that describes the service details, documents service level targets, and specifies the responsibilities of the Trusted System and the ISI.

6.6.2. ORGANIZATIONAL ROLES AND RESPONSIBILITIES

MISE Management

The MISE Management is responsible for the day-to-day operations for the MISE. The MISE Management provides operational service management functions for the MISE and serves as the primary point of contact for the MISE Help Desk. The following list includes some specific responsibilities and functions of the MISE Management:

- Certify and approve new trusted systems to participate in the environment
- Manages MISE CA and Trust Fabric Document
- Responsible for change and configuration management of the MISE
- Manages the ISI
- Defines operational process based on best practices to be documented in the MISE Operational Policies and Procedures

Trusted Systems

The following list includes specific responsibilities of *ALL* Trusted Systems:

- Adopt entitlement attributes
- Protect PII
- Notify MISE Management of any changes to security policy or posture
- Trust MISE Root CA
- Prepare SAML metadata for inclusion in trust fabric

Responsibilities specific to Information Providers include:

- Implement entitlement attributes or delegate enforcement to MISE
- Maintain information access policy based on entitlement attributes within MISE
- Implement publication, or search/retrieve

Responsibilities specific to Information Consumers include:

- Vet end users for access to the environment
- Provide authentication credentials to end users
- Authenticate end users
- Generate SAML user assertions containing entitlement attributes
- Implement search/retrieve

6.6.3. END USER SUPPORT

The MISE is comprised of many member organizations, each with its own local help desk resources, it is necessary to leverage these local resources to the maximum extent possible. The primary guiding principle in the design of the help desk structure is that all problems should be solved as close to the user as possible, and with as little centralized effort as possible; however, more complex technical issue not contained to a single trusted system will require a central help desk entity. The following three-tier help desk structure and issue escalation plan provides guidelines for the MISE.

Tier 1: Local Help Desk Support

All issues encountered by users should be first reported to the users local help desk. This level of user assistance is provided by the users local department to resolve simple issues reported by users (e.g., network outages, firewall problems, and local desktop user interface issues) should be handled at this level without bringing any higher-level resources into play.

Tier 2: Trusted System Help Desk Support

For any issue that the local help desk cannot resolve, the trusted system provides help desk support to the user. Issues that can be solved at Tier 2 may include questions regarding permissions and access control policies for information resources provisioned via the ISI or application-specific issues on the information consumer system. The trusted system help desk should attempt to resolve the issue, and will contact the MISE help desk support staff if the issue relates to an information provider or the ISI.

Tier 3: MISE Help Desk Support

Any issue that cannot be resolved at Tier 2 should be escalated to the MISE Help Desk. Issues that can be solved at Tier 3 include repairing a corrupted trust fabric document or resolving a technical issue that arise between multiple trusted systems. Issues that are resolved at this level will be tracked in the MISE Management's issue tracking database.

Appendices

A. Implementation Guide

B. Attribute Specification

C. Interface Security Specification

D. Publication Interface Specification

E. Search/Retrieve Specification

F. Governance Manual

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A - IMPLEMENTATION GUIDE

This implementation guide is copied from the online implementation guide at http://mise.mda.gov. Refer to the web site for the latest version.

The Implementation Guide contains the following Sections:

1. Introduction

Maritime security is a national priority that depends on the ability to efficiently, effectively, and appropriately share and safeguard information among trusted maritime partners within the Global Maritime Community of Interest (GMCOI). The Maritime Information Sharing Environment (MISE) as defined in the National Maritime Architecture Plan enables secure, standardized sharing of unclassified maritime information among a wide variety of federal, state and local agencies as well as international participants. MISE employs NIEM-M exchange models, representational state transfer (REST) services for publishing/consuming, and attribute-based access control to facilitate information sharing and safeguarding with non-provisioned users in a dynamic environment.

The purpose of this implementation guide is to provide practitioners with guidance and specific examples for interfacing with MISE. Specifically, it shows how to create messages that conform to the National Information Exchange Model (NIEM) Maritime IEPD formats, how to implement security to successfully access the environment, and how to interface with the services to publish and consume messages from the environment.

For new practitioners it is recommended you start with Process Flows for Security, Publish/Update, Delete, Search, and Retrieve and proceed in order through the implementation guide.

1.1. NIEM-M EXCHANGE MODELS

To learn more about using the NIEM-M exchange models, where they can be downloaded, and how to produce messages to adhere these standards review the section on Data Mapping.

1.2. SERVICE INTERFACES

To learn more about the service interfaces to share information via the MISE start with the section on Process Flows for Security, Publish/Update, Delete, Search, and Retrieve, and then visit the sections on Interfacing with the Publication Service , Interfacing with the Delete Service, Interfacing with the Search Service, and Interfacing with the Retrieve Service.

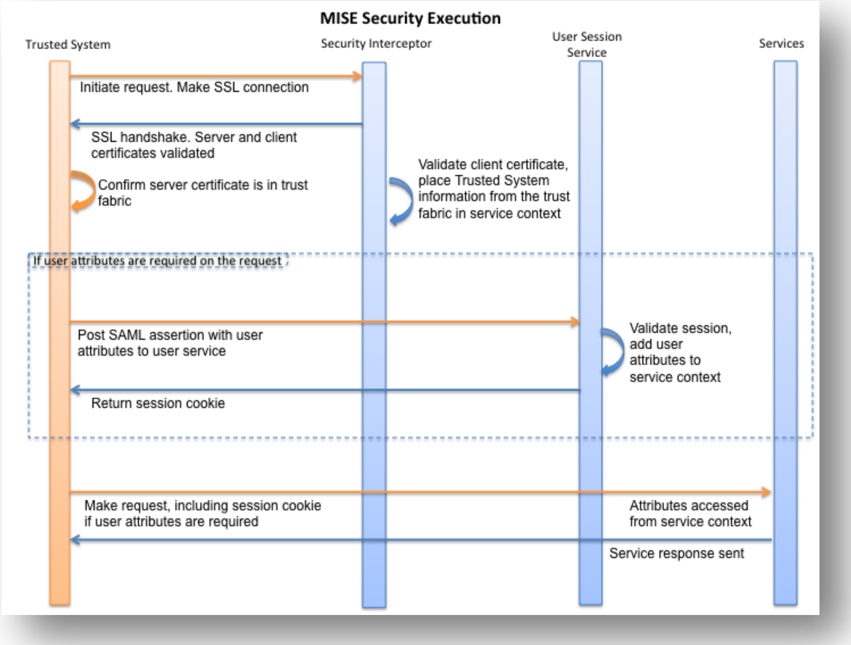
1.3. SECURITY SERVICES

To learn more about interfacing with the MISE security services see the sections on Process Flows for Security, Publish/Update, Delete, Search, and Retrieve and Interfacing with the Security Services.

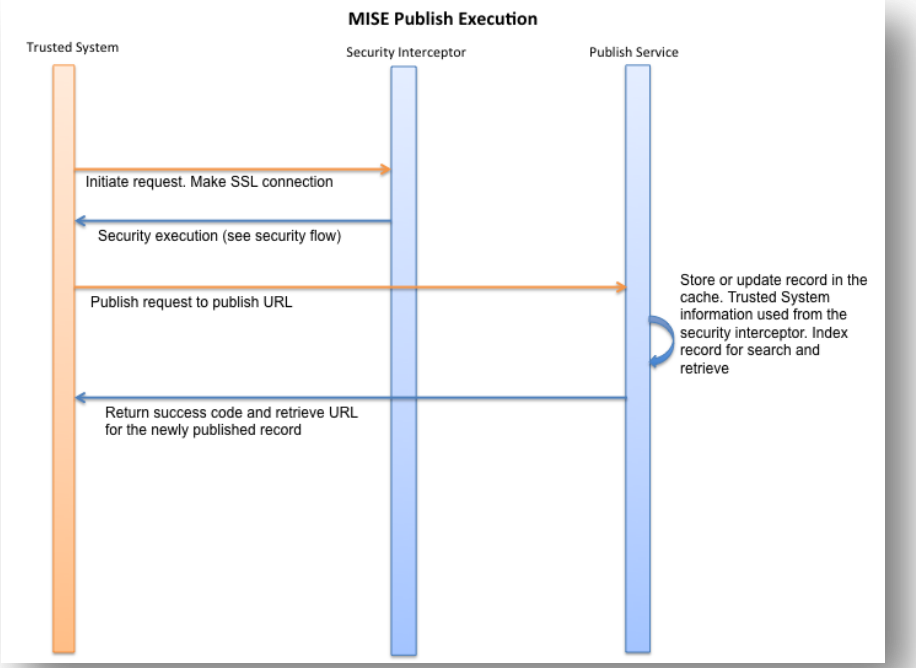
2. Process Flows for Security, Publish/Update, Delete, Search, and Retrieve

This section shows graphical representations of the major data provider and data consumer interactions with the MISE services.

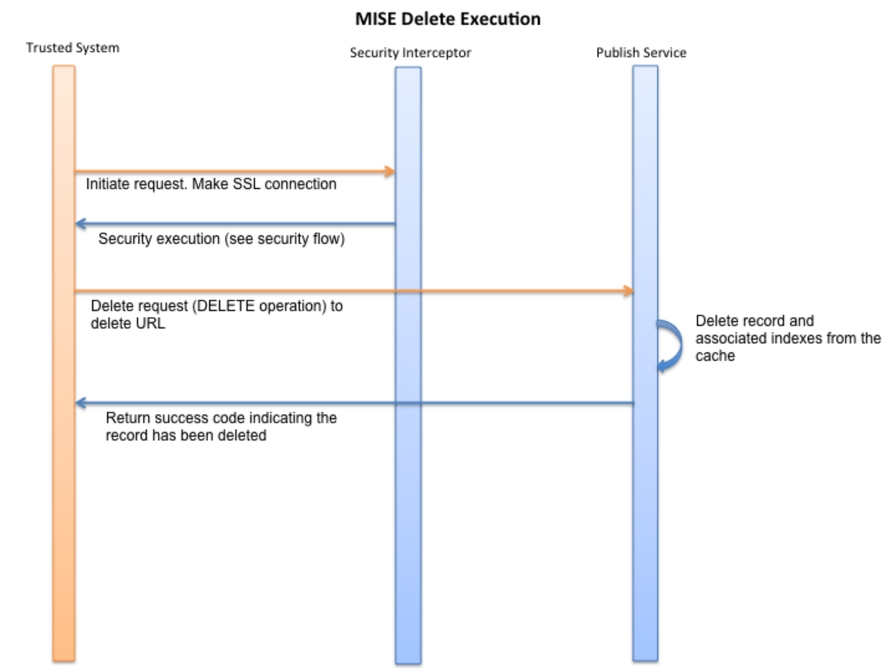
2.1. SECURITY


MISE Security Execution
Trusted System
Security Interceptor
User Session Service
Services
Initiate request. Make SSL connection
SSL handshake. Server and client certificates validated
Confirm server certificate is in trust fabric
Validate client certificate, place Trusted System information from the trust fabric in service context
If user attributes are required on the request
Post SAML assertion with user attributes to user service
Validate session, add user attributes to service context
Return session cookie
Make request, including session cookie if user attributes are required
Attributes accessed from service context
Service response sent

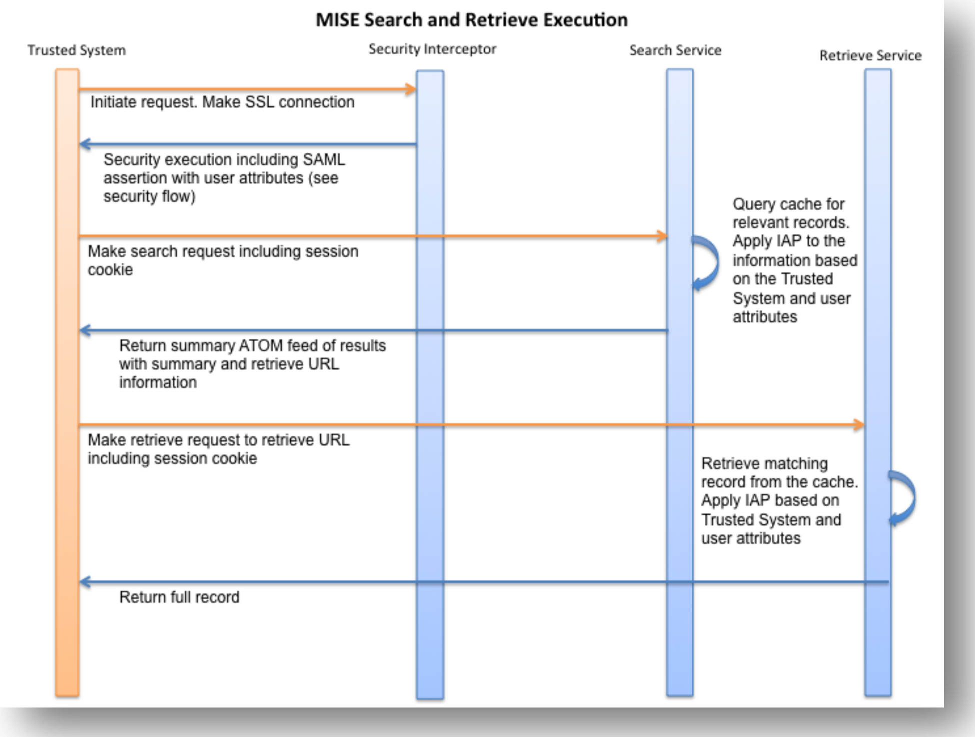
2.2. PUBLISH/UPDATE


MISE Publish Execution
Trusted System
Security Interceptor
Publish Service
Initiate request. Make SSL connection
Security execution (see security flow)
Publish request to publish URL
Store or update record in the cache. Trusted System information used from the security interceptor. Index record for search and retrieve
Return success code and retrieve URL for the newly published record

2.3. DELETE


MISE Delete Execution
Trusted System
Security Interceptor
Publish Service
Initiate request. Make SSL connection
Security execution (see security flow)
Delete request (DELETE operation) to delete URL
Delete record and associated indexes from the cache
Return success code indicating the record has been deleted

2.4. SEARCH AND RETRIEVE


MISE Search and Retrieve Execution
Trusted System
Security Interceptor
Search Service
Retrieve Service
Initiate request. Make SSL connection
Security execution including SAML assertion with user attributes (see security flow)
Make search request including session cookie
Query cache for relevant records. Apply IAP to the information based on the Trusted System and user attributes
Return summary ATOM feed of results with summary and retrieve URL information
Make retrieve request to retrieve URL including session cookie
Retrieve matching record from the cache. Apply IAP based on Trusted System and user attributes
Return full record

3. Data Mapping

3.1. OBTAIN THE LATEST NIEM-M MODELS

The NIEM-M Maritime Domain Awareness (MDA) Enterprise Information Exchange Model (EIEM) and Information Exchange Package Documentation (IEPD) are registered and available for download from the NIEM IEPD Clearinghouse and the DoD Metadata Registry.

Quick links to download the artifacts:

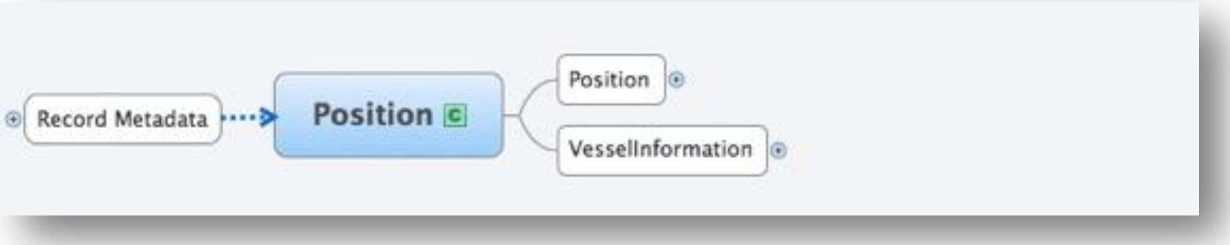
- Download MDA Enterprise Information Exchange Model (EIEM)
- Download Notices of Arrival IEPD
- Download Indicators and Notifications IEPD
- Download Vessel Positions IEPD

3.2. HOW TO MAP DATA TO NIEM MARITIME

Suppose the following is a native vessel position report format. Data elements include ship's identification, GPS position, course, speed, navigational status, and time-stamp. A sample XML syntax is included below.

```
<Track>
    <DateTime>2011-01-13T13:17:02.325Z</DateTime>
    <CountryMID>303</CountryMID>
    <RegionID>27</RegionID>
    <TrackInfo>
        <TrackInfoVessel>
            <VesselId>
                <MMSI>367354000</MMSI>
            </VesselId>
            <VesselAIS>
                <VesselDataDynamic>
                    <Coordinate>
                        <LAT>53.8782833</LAT>
                        <LON>-166.538633</LON>
                    </Coordinate>
                    <COG>178</COG>
                    <SOG>0.1</SOG>
                    <HDT>228</HDT>
                    <ROT>0</ROT>
                    <NavStatus>Engine</NavStatus>
                    <PosAcc>Low</PosAcc>
                </VesselDataDynamic>
            </VesselAIS>
        </TrackInfoVessel>
    </TrackInfo>
</Track>
```

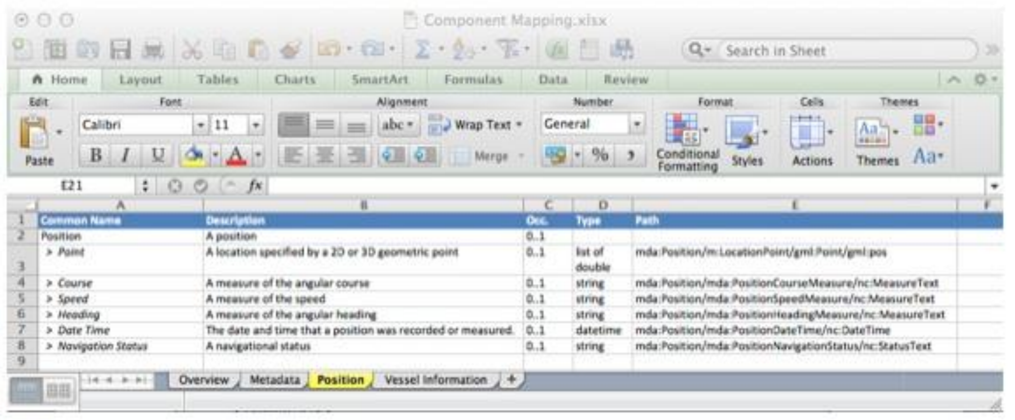
As depicted in the logical diagram for the Position IEPD, Position and Vessel Information are the two primary logical blocks included in a position report. Record Metadata is included in every message type.



The first step is to use the Mapping Spreadsheet from the IEPD. In this example, we are translating a track message into the NIEM format so we will be using the Position IEPD. Find and open the component mapping spreadsheet.

The mapping spreadsheet provides the path for each element in the XML. Each tab corresponds to a "block" in the logical diagram.

An example of the Mapping Spreadsheet is captured below:



Common Name	Description	Occ.	Type	Path
Position	A position	0..1		
> *Point*	A location specified by a 2D or 3D geometric point	0..1	list of double	mda:Position/m:LocationPoint/gml:Point/gml:pos
> *Course*	A measure of the angular course	0..1	string	mda:Position/mda:PositionCourseMeasure/nc:MeasureText
> *Speed*	A measure of the speed	0..1	string	mda:Position/mda:PositionSpeedMeasure/nc:MeasureText
> *Heading*	A measure of the angular heading	0..1	string	mda:Position/mda:PositionHeadingMeasure/nc:MeasureText
> *Date Time*	The date and time that a position was recorded or measured.	0..1	datetime	mda:Position/mda:PositionDateTime/nc:DateTime
> *Navigation Status*	A navigational status	0..1	string	mda:Position/mda:PositionNavigationStatus/nc:StatusText

Use the Position tab of the Mapping spreadsheet to determine the xml elements to represent the position data in the position message type. For the example message the position is represented as follows:

```
<mda:Position>
    <m:LocationPoint>
        <gml:Point gml:id="tp1">
            <gml:pos>53.8782833 -166.538633</gml:pos>
        </gml:Point>
    </m:LocationPoint>
    <mda:PositionSpeedMeasure>
        <nc:MeasureText>0.1</nc:MeasureText>
        <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
    </mda:PositionSpeedMeasure>
    <mda:PositionCourseMeasure>
        <nc:MeasureText>178</nc:MeasureText>
        <m:AngleUnitText>deg</m:AngleUnitText>
    </mda:PositionCourseMeasure>
    <mda:PositionHeadingMeasure>
        <nc:MeasureText>228</nc:MeasureText>
        <m:AngleUnitText>deg</m:AngleUnitText>
    </mda:PositionHeadingMeasure>
    <mda:PositionNavigationStatus>
        <nc:StatusText>Engine</nc:StatusText>
    </mda:PositionNavigationStatus>
    <mda:PositionDateTime>
        <nc:DateTime>2011-01-13T13:17:02.325Z</nc:DateTime>
    </mda:PositionDateTime>
</mda:Position>
```

If an element does not map to the NIEM format, it can be included in the expansion text if desired.

To complete the message, we used the Vessel Information tab of the Mapping Spreadsheet to complete translation. Vessel is represented as follows:

```
<mda:Vessel>
    <m:VesselAugmentation>
        <m:VesselMMSIText>367354000</m:VesselMMSIText>
    </m:VesselAugmentation>
</mda:Vessel>
```

Below is the full NIEM conformant Position Message.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample Position instance corresponding to the Position version 3.2 IEPD -->
<posex:Message
    xsi:schemaLocation="<a href="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2">http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2</a> ../XMLSchemas/exchange/3.2/position-exchange.xsd"
    xmlns:m="<a href="http://niem.gov/niem/domains/maritime/2.1">http://niem.gov/niem/domains/maritime/2.1</a>" xmlns:mda="<a href="http://niem.gov/niem/domains/maritime/2.1/mda/3.2">http://niem.gov/niem/domains/maritime/2.1/mda/3.2</a>"
    xmlns:posex="<a href="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2">http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2</a>"
    xmlns:nc="<a href="http://niem.gov/niem/niem-core/2.0">http://niem.gov/niem/niem-core/2.0</a>" xmlns:gml="<a href="http://www.opengis.net/gml/3.2">http://www.opengis.net/gml/3.2</a>"
    xmlns:ism="urn:us:gov:ic:ism" xmlns:xsi="<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>"
    mda:securityIndicatorText="LEI" mda:releasableNationsCode="USA"
    mda:releasableIndicator="true">
    <nc:DocumentCreationDate>
        <nc:Date>2011-12-01</nc:Date>
    </nc:DocumentCreationDate>
    <nc:DocumentExpirationDate>
        <nc:Date>2012-01-01</nc:Date>
    </nc:DocumentExpirationDate>
    <nc:DocumentCreator>
        <nc:EntityOrganization>
            <nc:OrganizationName>Example Organization</nc:OrganizationName>
        </nc:EntityOrganization>
    </nc:DocumentCreator>
    <mda:RecordIDURI>00000001</mda:RecordIDURI>
    <mda:MessageStatusCode>Initial</mda:MessageStatusCode>
    <mda:MessageSourceSystemName>Track Source</mda:MessageSourceSystemName>
    <mda:ICISMMarkings ism:classification="U"
        ism:ownerProducer="USA" />
    <mda:Vessel>
        <m:VesselAugmentation>
            <m:VesselMMSIText>367354000</m:VesselMMSIText>
        </m:VesselAugmentation>
    </mda:Vessel>
    <mda:Position>
        <m:LocationPoint>
            <gml:Point gml:id="tp1">
                <gml:pos>53.8782833 -166.538633</gml:pos>
            </gml:Point>
        </m:LocationPoint>
        <mda:PositionSpeedMeasure>
            <nc:MeasureText>0.1</nc:MeasureText>
            <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
        </mda:PositionSpeedMeasure>
        <mda:PositionCourseMeasure>
            <nc:MeasureText>178</nc:MeasureText>
            <m:AngleUnitText>deg</m:AngleUnitText>
        </mda:PositionCourseMeasure>
        <mda:PositionHeadingMeasure>
            <nc:MeasureText>228</nc:MeasureText>
            <m:AngleUnitText>deg</m:AngleUnitText>
        </mda:PositionHeadingMeasure>
        <mda:PositionNavigationStatus>
            <nc:StatusText>Engine</nc:StatusText>
        </mda:PositionNavigationStatus>
        <mda:PositionDateTime>
            <nc:DateTime>2011-01-13T13:17:02.325Z</nc:DateTime>
        </mda:PositionDateTime>
    </mda:Position>
</posex:Message>
```

4. Code Overview

The MISE implementation team provides a client toolkit for interface to the MISE REST services for Publish, Update, Delete, Search, and Retrieve. All operations are simply REST operations to the correct endpoint.

The client toolkit is primarily designed to make interfacing with the security services easier. The client toolkit is implemented in accordance with the following specifications. More information about each of the specifications can be found in the links below.

- National MDA Architecture Attribute Specification
- National MDA Architecture Security Specification
- National MDA Architecture Publish Specification
- National MDA Architecture Search/Retrieve Specification

The client toolkit can be downloaded from the MDA Architecture tools page. The tools contain a simple java project that demonstrates how to connect and make a GET request against the ISI services. Two JAR files are included. The first is the MDAUtils JAR, which contains the MDA Architecture security implementation, the client REST toolkit, and all the necessary dependencies. Additionally, the project also requires the included commons-io JAR, which provides file handling utilities for reading and writing files.

The client toolkit is used in all of the following examples:

- Interfacing with the Security Services
- Interfacing with the Publication Service
- Interfacing with the Delete Service
- Interfacing with the Search Service
- Interfacing with the Retrieve Service

Please note that the current base URL for the MISE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

5. User Stories for Search

Prior to reading this section, read Process Flows for Security, Update, Delete, Search, and Retrieve for a basic understanding of how information flows between the provider and consumer.

This set of user stories calls out specific examples for search and retrieve for the information products provided by the MISE. Examples for publish and delete are contained in the sections that discuss those operations, as they are simple, one-time operations.

As noted in the National MDA Search/Retrieve Specification, these operations return the Atom summary feeds of each of the information products. The full message for any of the summaries would be accessed via a retrieve operation.

in each of the examples below, the URL is relative to the mise.mda.gov base path for MISE service access. The placeholder $value is used in the place of query values.

5.1. POSITION

Return vessel position summary messages based on a geospatial area and time window, to see last known positions of all vessels within that geospatial area.	/search/pos?ulat=$value&ulng=$value&llat=$value&llng=$value&start=$value&end=$value
Retrieve a full vessel position message from the URL in a position summary for full details on a specific vessel.	/retrieve/pos?entityid=$eid&recordid=$posid
Retrieve the most recent vessel position summary data for each vessel that has updated in the last 30 seconds in the geographic area of interest.	/search/pos?ulat=$value&ulng=$value&llat=$value&llng=$value&start=$value&end=$value start, end should be the last 30 seconds

5.2. INDICATORS AND NOTIFICATIONS

Retrieve a summary of IANs about all vessels in a specific geospatial area.	/search/ian?ulat=$value&ulng=$value&llat=$value&llng=$value &start=$value&end=$value
Retrieve a summary message for vessels with a specific threat level within a geospatial area. Note that $threat in the following example must align with one of the threat values available in the IAN schema.	/search/ian?ulat=$value&ulng=$value&llat=$value&llng=$value&$threat=$value

5.3. NOTICE OF ARRIVAL

Retrieve a summary message of all vessels inbound to a specified port.	/search/noa?PortCodeText=$value&start=$value&end=$value
Retrieve a summary message of all pending notices of arrival for a specified vessel.	/search/noa?VesselNameText=$value&VesselMMSIText=$value &VesselIMONumberText=$value
Retrieve a summary of IANs of all vessels in a specific geospatial area that are carrying certain dangerous cargo (CDC).	/search/ian?ulat=$value&ulng=$value&llat=$value&llng=$value &start=$value&end=$value&VesselCDCCargoOnboardIndicator=true
Retrieve a summary message based on a specified vessel and port for which any data element has been updated in	/search/noa?start=$value&end=$value&PortCodeText=$value&VesselNameText=$value

the last 10 minutes.	&VesselMMSIText=$value&VesselIMONumberText=$value start, end should be the last 10 minutes

*Note that any combination of MMSI, IMO, and Name can be used, all three are not required.

5.4. LEVELS OF AWARENESS

Retrieve a summary message for all vessels in a specified geospatial area.	/search/loa?ulat=$value&ulng=$value&llat=$value&llng=$value&start=$value&end=$value
Retrieve a summary message for vessels with a specific threat level within a geospatial area. Note that $threat in the following example must align with one of the threat values available in the LOA schema.	/search/loa?ulat=$value&ulng=$value&llat=$value&llng=$value&$threat=$value
Retrieve a summary of a specific vessel in a geospatial area.	/search/loa?ulat=$value&ulng=$value&llat=$value&llng=$value &VesselNameText=$value&VesselMMSIText=$value&VesselIMONumberText=$value

*Note that any combination of MMSI, IMO, and Name can be used, all three are not required.

6. Interfacing with the Security Services

All interactions to publish and consume data within the MISE are secured interactions over SSL between trusted systems. As a prerequisite to understanding the security implementation examples in this section, it is highly recommend you first read the following documents:

- National MDA Architecture Plan for an overview of the MISE security approach.
- National MDA Architecture Security Specification for the details of how trusted systems securely connect to the ISI.
- National MDA Architecture Attribute Specification for an explanation of the common attributes used for entitlement management.

6.1. OBTAINING X.509 CERTIFICATES

Numerous tools and processes are available for creating key pairs and X.509 certificates. The exact process chosen by a trusted system will vary depending on the platform the trusted system implementation is based upon, agency procedures, and the chosen root CA.

In some cases a trusted system may need to generate a keypair and a certificate signing request (CSR) internally using a tool such as OpenSSL or Java's keytool, and submit the CSR to a root CA for signing. The following sections provide steps for generating the private key and public Certificate Signing Request (CSR).

6.2. USING OPENSSL TO GENERATE A PRIVATE KEY AND PUBLIC CERTIFICATE SIGNING REQUEST (CSR)

Issue the following command to create private key and CSR

```
openssl req -new -nodes -keyout myserver.key -out server.csr -newkey rsa:2048
```

This creates two files. The file myserver.key contains a private key; do not disclose this file to anyone. Carefully protect the private key. In particular, be sure to back up the private key, as there is no means to recover it should it be lost. The private key is used as input in the command to generate a Certificate Signing Request (CSR).

1. You will now be asked to enter details to be entered into your CSR.
   What you are about to enter is what is called a Distinguished Name or a DN. Use the FQDN as Common Name (CN). The fields email address, optional company name and challenge password can be left blank for your SSL certificate.

```
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:California
Locality Name (eg, city) []:San Diego
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Agency One
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:agencyone.gov
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
-----
```

2. Your CSR has been created. You can open the server.csr in a text editor to view it.

3. Follow the CA-specific instructions for submitting the CSR to the CA to generate your SSL certificate.

6.3. Use Java's keytool to generate a private key and public Certificate Signing Request (CSR)

1. Using the java keytool command line utility, the first thing you need to do is create a key store and generate the key pair. Do this with the following command:

```
keytool -genkey -keysize 2048 -keyalg RSA -alias agencyone -keystore
mykeystore
```

4. Tip: The 2048 in the command above is the key bit length. MISE requires a key bit length of 2048.

5. You will be prompted for a password for the key store. The key password must be at least 6 characters long.

6. Tip: Make a note of the password. If lost it cannot be retrieved.

7. You will be asked for several pieces of info which will be used by the MISE Test Certificate Authority to create your new SSL certificate. When it asks for your first and last name, make sure you enter the FQDN of your server that will make the connection to the MISE. Here is an example:

```
What is your first and last name?
[Unknown]: http://agencyone.mda.gov
What is the name of your organizational unit?
[Unknown]: IT
```

```
What is the name of your organization?
[Unknown]: Agency One
What is the name of your City or Locality?
[Unknown]: San Diego
What is the name of your State or Province?
[Unknown]: California
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=http://agencyone.mda.gov, OU=IT, O=Agency One, L=San Diego,
ST=California, C=US correct?
[no]: yes
```

8. You will be prompted for a password for the private key within the key store. If you press enter at the prompt, the key password is set to the same password as that used for the key store from the previous step. The key password must be at least 6 characters long.
9. Tip: Make a note of the passwords. If lost they cannot be retrieved.
10. Now generate the Certificate Signing Request (CSR) from the private key generated above using the following command:

    ```
    keytool -certreq -alias agencyone -file agencyone.gov.csr -keystore mykeystore
    ```

    This creates a CSR and stores it in a file named agencyone.gov.csr.
11. Follow the CA-specific instructions for submitting the CSR to the CA to generate your SSL certificate.

Below is an example of what your CSR will look like. This is an example only and cannot be used to generate your SSL certificate.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIICtTCCAZ0CAQAwcDELMAkGA1UEBhMCVVMxEzARBgNVBAgTCmNhbGlmb3JuaWExEjAQBgNVBAcT
CXNhbiBkaWVnbzETMBEGA1UEChMKYWdlbmN5IG9uZTELMAkGA1UECxMCSVQxFjAUBgNVBAMTDWFn
ZW5jeW9uZS5nb3YwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCwjMqyx0R2OqhltjXJ
4tH7275YW01+6K6kLOc/yFmfDLK8oSynDoTq4PzO2z1BAhq/8CgkPTs/tHgNphWTlw0c5WpaR487
bVh0dyJwvz3EI6hLmPAyqTvAB2C2aW0zcLzGTSxR8rAhHoX7oOgA3E9xKmoYMVMIZLFN63Tn/F6M
T5NdFdTbkoRzcxpkkVmH6o60Vv6jGTI+zUpdyC7W8QRm/kshQgtjXLeYLACXuLvaKzn69p1TLCss
knRCsOsLjHhJrBmPK3upD9HRZ2bbE+Pp1/QUUVztiHDhnwPyEV6Iq2ZFOAuIF4otQU05DLQMsI28
EPu52GmfDMkPuXZFmYYDAgMBAAGgADANBgkqhkiG9w0BAQUFAAOCAQEAJTwTApJiggCgSxE48+Wi
ATNHe3rHJYPLzFMtRupM0tReLQWA+246g+ZGFHOwRv2VO90mMW/MivoxAnoyyP5J708MNsHo1LMn
bW9kyUuZK22T0lpO3t6BDDix8NWLg5cEGm08sI20iptesemlKq4E/2TxFdEmLpiARD9768WGVTbX
8EB08V2U78A8s5hTMly7hnaywfOm4ezpWllktUlEuzVxGLHkBj7H5CEKPjH02/AZRNYJRYWrzcdO
YS/gUqs/cvqL77QrwOXWjrCEjSKYtibaXNlSbjEnDKbkoKJl0UsKRLAhMs8NI/HvalV1o8J8/ftc
1J1xTgHFYyxRJluV6w==
-----END NEW CERTIFICATE REQUEST-----
```

6.4. REGISTRATION OF TRUSTED SYSTEM IN TRUST FABRIC

Once the necessary X.509 certificate is obtained, your trusted system must be registered in the trust fabric document by the MISE Management team. You will have an entry in the trust fabric for each role for which your system is authorized, i.e. data provider and/or data consumer.

Following is an example entry for a provider trusted system:

```
<md:EntityDescriptor entityID="<a
href="https://mise.agencythree.gov/">https://mise.agencythree.gov/</a>">
    <md:RoleDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
        xsi:type="mise:MISEProviderDescriptorType">
        <md:KeyDescriptor use="signing">
```

```
            <ds:KeyInfo xmlns:ds="<a
href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>">
                <ds:X509Data>
                    <ds:X509Certificate>
                        <!-- Base 64 encoded certificate embedded here
                        This is the client certificate which the trusted
                        system will present during SSL connection handshake.
                        The private key matching this certificate will also
                        be used by this trusted system for signing SAML
                        assertions.
                        -->
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>
    </md:RoleDescriptor>
    <md:ContactPerson contactType="technical">
        <md:Company>Trusted Federal Systems, Inc.</md:Company>
        <md:GivenName>Eric</md:GivenName>
        <md:SurName>Jakstadt</md:SurName>
        <md:EmailAddress><a
href="mailto:eric.jakstadt@trustedfederal.com">eric.jakstadt@trustedfederal.com</a
></md:EmailAddress>
        <md:TelephoneNumber>404-806-8143</md:TelephoneNumber>
    </md:ContactPerson>
</md:EntityDescriptor>
```

Now an example entry for a consumer trusted system. Notice in the consuming system entry in the trust fabric, the trusted system is assigned the appropriate indicator attributes used to make authorization decisions on queries.

```
<md:EntityDescriptor entityID="<a
href="https://mise.agencyone.gov/">https://mise.agencyone.gov/</a>">
    <md:Extensions>
        <gfipm:EntityAttribute FriendlyName="COIIndicator"
            Name="mise:1.2:entity:COIIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue
xsi:type="xs:string">True</gfipm:EntityAttributeValue>
        </gfipm:EntityAttribute>
        <gfipm:EntityAttribute FriendlyName="LawEnforcementIndicator"
            Name="mise:1.2:entity:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue
xsi:type="xs:string">True</gfipm:EntityAttributeValue>
        </gfipm:EntityAttribute>
        <gfipm:EntityAttribute FriendlyName="PrivacyProtectedIndicator"
            Name="mise:1.2:entity:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue
xsi:type="xs:string">True</gfipm:EntityAttributeValue>
        </gfipm:EntityAttribute>
        <gfipm:EntityAttribute FriendlyName="OwnerAgencyCountryCode"
            Name="mise:1.2:entity:OwnerAgencyCountryCode"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue
xsi:type="xs:string">USA</gfipm:EntityAttributeValue>
        </gfipm:EntityAttribute>
    </md:Extensions>
    <md:RoleDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
```

```
        xsi:type="mise:MISEConsumerDescriptorType">
        <md:KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="<a
href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>">
                <ds:X509Data>
                    <ds:X509Certificate>
                        <!-- Base 64 encoded certificate embedded here
                                 This is the client certificate which the trusted
                                system will present during SSL connection
handshake.
                                The private key matching this certificate will
also
                                be used by this trusted system for signing SAML
                                assertions.
                                -->
                    </ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>
    </md:RoleDescriptor>
    <md:Organization>
        <md:OrganizationName xml:lang="en">Agency One</md:OrganizationName>
        <md:OrganizationDisplayName xml:lang="en">Agency
            One</md:OrganizationDisplayName>
        <md:OrganizationURL xml:lang="en"><a
href="http://www.agencyone.gov/</md:OrganizationURL">http://www.agencyone.gov/</md
:OrganizationURL</a>>
    </md:Organization>
    <md:ContactPerson contactType="technical">
        <md:Company>Trusted Federal Systems, Inc.</md:Company>
        <md:GivenName>Eric</md:GivenName>
        <md:SurName>Jakstadt</md:SurName>
        <md:EmailAddress><a
href="mailto:eric.jakstadt@trustedfederal.com">eric.jakstadt@trustedfederal.com</a
></md:EmailAddress>
        <md:TelephoneNumber>404-806-8143</md:TelephoneNumber>
    </md:ContactPerson>
</md:EntityDescriptor>
```

6.5. Download the Trust Fabric Document

As discussed in the Security Specification, the trust fabric contains public keys for all trusted systems that interact with the MISE. The trust fabric endpoint requires HTTPS but does not require a client certificate or any other method of authentication.

Retrieve the trust fabric document by any standard means, including viewing in any browser, at the MISE server at /miseresources/TrustFabric.xml

The trust fabric document for the MISE test environment is available at: https://107.23.66.168:9443/miseresources/TrustFabric.xml

6.6. Validate the Trust Fabric Signature Programmatically

The following sample code (written in Java) taken from client toolkit demonstrates validating of the signed trust fabric document. Since the trust fabric document is a SAML metadata file with a few simple extensions, this sample code is able to leverage the open source OpenSAML project to simplify implementation. Trusted system implementations not written in Java, or which already include other SAML implementations, may also be able to simplify implementation by relying on existing SAML metadata implementations.

The following code snippet shows how the trust fabric document may be loaded into a DOM object so the signing certificate can be parsed and the signature on the document validated.

```
    // read in the trust fabric from a local file location
    FileInputStream fis = new FileInputStream("/local/path/TrustFabric.xml");
    m_domFactory = DocumentBuilderFactory.newInstance();
    m_domFactory.setNamespaceAware(true);
Element domElement =
m_domFactory.newDocumentBuilder().parse(fis).getDocumentElement();

// cryptographic validation of signature
X509Certificate signedByCert = verifyXMLSignature(domElement);
System.out.println(String.format("Signature validation %s", signedByCert == null ?
"FAILED" : "SUCCEEDED"));
```

The following snippet takes the trust fabric as a DOM object and returns the signing certificate if it is valid.

```
public static X509Certificate verifyXMLSignature(Element target) throws Exception {
    // Validate the signature -- i.e. SAML object is pristine:
    NodeList nl = target.getElementsByTagNameNS(XMLSignature.XMLNS, "Signature");
    if (nl.getLength() == 0)
        return null;

    KeyValueKeySelector kvs = new KeyValueKeySelector();
    DOMValidateContext context = new DOMValidateContext(kvs, nl.item(0));

    // Create a DOM XMLSignatureFactory that will be used to unmarshal the
    // document containing the XMLSignature
    String providerName = System.getProperty("jsr105Provider",
"org.jcp.xml.dsig.internal.dom.XMLDSigRI");
    XMLSignatureFactory fac = XMLSignatureFactory.getInstance("DOM", (Provider)
Class.forName(providerName).newInstance());

    DOMXMLSignature signature = (DOMXMLSignature)
fac.unmarshalXMLSignature(context);
    if (!signature.validate(context))
        return null;

    return kvs.getUsedCertificate();
}
```

6.7. Implementing MISE Security Attributes

As detailed in the National MDA Attribute Specification , entitlement management within the MISE relies on the use of a common set of entity, user, and data attributes to make run-time authorization decisions as to whether a trusted system and requesting user are authorized to access a requested information resource.

There are three categories for attributes defined for the National MDA Architecture:

1. Entity Attributes: Attributes that pertain to a trusted system within the MISE.
2. User Attributes: Attributes that pertain to a human user.
3. Data Attributes: Attributes that pertain to data.

Currently data is grouped by LE sensitive (LEI), privacy protected (PPI) and the rest of the community (COI). The security indicators defined in the attribute specification map to these groups, i.e. Law Enforcement Indicator, Privacy Protected Indicator, and COI Indicator. Additionally, there is a one-to-one relationship between the security indicators assigned to data (data attributes) by information providers to convey sharing restrictions and the indicators assigned to information consumer trusted systems (entity attributes) and users (user attributes) to convey their respective privileges.

6.8. APPLYING DATA ATTRIBUTES ON PUBLISH

As a Data Provider Trusted System you must tag messages before publishing to the ISI with metadata to convey any restrictions on the data.

Attribute Name	Possible Values	Description
securityIndicatorText	"LEI" | "PPI" | "COI"	Indicates the level of access required to access the data. LEI for Law Enforcement sensitive information, PPI for privacy protected information, or COI for the rest of the community.
releasableIndicator	"true" | "false"	Marks data as releasable to the public domain under the restrictions of the associated security indicator.
releasableNationsCode	Space-delimited list of 3-letter country codes, ex. "CAN USA FRA"	Indicates data can only be released to those nations identified by the country codes. Default value is "USA".

For example to publish a vessel position messages that is law enforcement sensitive, not publically releasable, and shareable with only US, these fields will be added to the exchange element of the publish message as depicted in the example below.

```
<message
xmlns:posex="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2"
mda:securityindicatortext="LEI" mda:releasablenationscode="USA"
mda:releasableindicator="false"></message>
```

6.9. SUPPLYING USER/ENTITY ATTRIBUTES FOR SEARCH/RETRIEVE

6.9.1. MAP LOCAL USER PRIVILEGES TO MISE SECURITY ATTRIBUTES

Use the MISE security attributes as defined in the Attribute Specification to assert citizenship and the access level for the user associated with a query. Citizenship is conveyed using the CitizenshipCode attribute, `mise:1.4:user:CitizenshipCode` , with a value equal to the ISO 3-letter country code.

The access level is conveyed using one or more attributes defined as indicators.

Indicator	**Attribute Name**	**Description**
Law Enforcement Indicator	mise:1.4:user:LawEnforcementIndicator	User requires and qualifies for access to law enforcement information in accordance with all appropriate statutes and legislation.

Privacy Protected Indicator	mise:1.4:user:PrivacyProtectedIndicator	User requires and qualifies for access to privacy protected information in accordance with all appropriate statutes and legislation.
Community of Interest Indicator	mise:1.4:user:COIIndicator	Minimum access level assigned to user that requires access to information shared by the MISE community.

6.9.2. FORMING SAML USER ASSERTION

The following code snippet provides an example of building the user assertions and adding them to the context of the request. The full example is shown in the section on Interfacing with the Search Services.

```
//Form the user assertion
            String assertingPartyID = "test.client";
            AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
            builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION,
10*60);  // valid for 10 minutes
            builder.addAttribute("ElectronicIdentityId",
"gfipm:2.0:user:ElectronicIdentityId", "<a
href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
            builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T.
User");

                        Attribute attr = builder.addAttribute("CitizenshipCode",
"mise:1.4:user:CitizenshipCode", "USA");
            builder.addAttribute("LawEnforcementIndicator",
"mise:1.4:user:LawEnforcementIndicator", "true");
            builder.addAttribute("PrivacyProtectedIndicator",
"mise:1.4:user:PrivacyProtectedIndicator", "true");

            builder.signUsingPkcs12(assertingPartyID,
FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
            Assertion assertion = builder.getAssertion();

            //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
cause the signature validation to fail
            HttpResponse response = m_client.post("/MDAUserSessionService/login",
null, SAMLUtils.asXMLString(assertion),
```

7. Interfacing with the Publication Service

The publication service provides the interface to publish information products to the MISE. The complete interface description for publish is in the Code Overview page for the code download and library details.

This example walks through publishing a single Position instance to the position interface, assuming that the publishing system can already interface with the security services, including registering the publishing system as a trusted system in the Trust Fabric.

The instance file that might be published via this operation can be downloaded here. This file contains a NIEM-M Position instance containing three position reports for the MV Example.

The actual XML for a publish operation would normally be assembled from a database or some other storage location. The example below just reads the XML from a file.

```
package test;
```

```

import gov.mda.trustfabric.TrustFabric;
import gov.mda.util.RestServiceClient;

import java.io.File;
import java.io.FileInputStream;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.xpath.XPathFactory;

import org.apache.commons.io.FileUtils;
import org.apache.commons.io.FilenameUtils;
import org.apache.http.HttpResponse;
import org.apache.http.entity.ContentType;

public class TestPublishClient {

    /**
     * @param args
     */
    public static void main(String[] args) {
        RestServiceClient m_client;


        /* Strongly recommend that these be loaded from a configuration file dynamically in production code */

        String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public certificate for the MISE
        String trustFabricUrl = "<a href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/miseresources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
        String trustFabricBackupPath = "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location for a cached version of the trust fabric
        String serverScheme = "https";
        String serverHost = "mise.mda.gov";
        String serverPort = "9443";
        String serverBasePath = "/services";
        String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore which contains the certificate and private key for this trusted system
        String keystorePass = "password";



        try {
            FileInputStream isCert = new FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
            CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
            X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);

                   TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);

            m_client = new RestServiceClient(serverScheme, serverHost, Integer.valueOf(serverPort), serverBasePath);

            m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath), keystorePass);

            String body = null;
            FileUtils.readFileToString(new File("SamplePosition.xml"), body);

            HttpResponse response = m_client.put("/publish/pos/id", "", body,
ContentType.APPLICATION_XML); //perform the request
        } catch(Exception e) {
            e.printStackTrace();
        }


    }

}
```

Note again that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```
FileInputStream isCert = new
FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);

TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);
```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```
m_client = new RestServiceClient(serverScheme, serverHost,
Integer.valueOf(serverPort), serverBasePath);

m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);
```

Finally, the last few lines read the XML from a file and publish it to the MISE. The /id on the publish URL must actually be a unique ID for this Position message in the publishing system.

```
String body = null;
FileUtils.readFileToString(new File("SamplePosition.xml"), body);

HttpResponse response = m_client.put("/publish/pos/id", "", body,
ContentType.APPLICATION_XML); //perform the request
```

In production, the publishing, response-handling, and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the publish operation.

Please note that the current base URL for the MSIE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

8. Interfacing with the Delete Service

The delete interface on the MISE is an extension of the publish interface. To delete a previously published information product, the publishing system need only issue a DELETE HTTP request with the same parameters as the original publish message. The complete interface description for delete is in the Publish Specification. See the Code Overview page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

The following code example is structurally identical to the publish example, save for the actual HTTP operation on line 47, which is a DELETE, instead of a PUT with XML content.

```
1   package test;
2
3   import gov.mda.trustfabric.TrustFabric;
4   import gov.mda.util.RestServiceClient;
5
6   import java.io.FileInputStream;
7   import java.security.cert.CertificateFactory;
8   import java.security.cert.X509Certificate;
9
10  import org.apache.commons.io.FilenameUtils;
11  import org.apache.http.HttpResponse;
12
13  public class TestDeleteClient {
14
15      /**
16       * @param args
17       */
18      public static void main(String[] args) {
19          RestServiceClient m_client;
20
21
22          /* Strongly recommend that these be loaded from a configuration file
    dynamically in production code */
23
24          String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
    certificate for the MISE
25          String trustFabricUrl = "<a
    href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/mise
    resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
26          String trustFabricBackupPath =
    "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
    for a cached version of the trust fabric
27          String serverScheme = "https";
28          String serverHost = "mise.mda.gov";
29          String serverPort = "9443";
30          String serverBasePath = "/services";
31          String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
    which contains the certificate and private key for this trusted system
32          String keystorePass = "password";
33
34
35
36          try {
37              FileInputStream isCert = new
    FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
38              CertificateFactory certFactory =
    CertificateFactory.getInstance("X.509");
39              X509Certificate cert = (X509Certificate)
certFactory.generateCertificate(isCert);

            TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath,
cert);

            m_client = new RestServiceClient(serverScheme,
serverHost,  Integer.valueOf(serverPort), serverBasePath);

            m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);

            HttpResponse response = m_client.delete("/publish/pos/id", "", null,
null);
        } catch(Exception e) {
            e.printStackTrace();
        }


    }

}
```

For the DELETE operation, the /id on the publish URL must actually be the ID under which the information product was originally published.

```
HttpResponse response = m_client.delete("/publish/pos/id", "", null, null);
```

In production, the delete, response-handling, and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the delete operation.

Please note that the current base URL for the MSIE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

9. Interfacing with the Search Service

The search service provides the interface to search for information products on the MISE. The complete interface description for search is in the Search/Retrieve Specification. See the Code Overview page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

This example walks through a query for Position reports based on the first Position User Story, a search bounded by geospatial area and time. This example assumes the searching system can interface with the MISE security services, including registering the system as a trusted system in the Trust Fabric.

Unlike the previous two examples which only require the SSL handshake with the MISE, the search and retrieve operations require that the consumer system pass the entitlement attributes of the user. The meanings of the attributes are discussed in detail in the Attribute Specification. These entitlement attributes are provided to the MISE via a SAML assertion. When the SAML assertion is validated, the MISE returns a session cookie to the client system, which must be provided for all subsequent search and retrieve operations.

```
package test;
```

```

import gov.mda.Constants;
import gov.mda.saml.AssertionBuilder;
import gov.mda.saml.SAMLUtils;
import gov.mda.trustfabric.TrustFabric;
import gov.mda.util.RestServiceClient;

import java.io.FileInputStream;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import org.apache.commons.io.FilenameUtils;
import org.apache.http.HttpResponse;
import org.apache.http.entity.ContentType;
import org.apache.http.util.EntityUtils;
import org.opensaml.saml2.core.Assertion;
import org.opensaml.saml2.core.Attribute;

public class TestSearchClient {

    /**
     * @param args
     */
    public static void main(String[] args) {
        RestServiceClient m_client;

        /* Strongly recommend that these be loaded from a configuration file
dynamically in production code */

        String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public
certificate for the MISE
        String trustFabricUrl = "<a
href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/mise
resources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
        String trustFabricBackupPath =
"C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location
for a cached version of the trust fabric
        String serverScheme = "https";
        String serverHost = "mise.mda.gov";
        String serverPort = "9443";
        String serverBasePath = "/services";
        String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore
which contains the certificate and private key for this trusted system
        String keystorePass = "password";

        try {
            FileInputStream isCert = new
FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
            CertificateFactory certFactory =
CertificateFactory.getInstance("X.509");
            X509Certificate cert = (X509Certificate)
certFactory.generateCertificate(isCert);

                    TrustFabric.initializeFromURL(trustFabricUrl,
trustFabricBackupPath, cert);

            m_client = new RestServiceClient(serverScheme,
serverHost,  Integer.valueOf(serverPort), serverBasePath);
            m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);

           //Form the user assertion
            String assertingPartyID = "test.client";
            AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
            builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION,
10*60);  // valid for 10 minutes
            builder.addAttribute("ElectronicIdentityId",
"gfipm:2.0:user:ElectronicIdentityId", "<a
href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
            builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T.
User");

                        Attribute attr = builder.addAttribute("CitizenshipCode",
"mise:1.4:user:CitizenshipCode", "USA");
            builder.addAttribute("LawEnforcementIndicator",
"mise:1.4:user:LawEnforcementIndicator", "true");
            builder.addAttribute("PrivacyProtectedIndicator",
"mise:1.4:user:PrivacyProtectedIndicator", "true");

            builder.signUsingPkcs12(assertingPartyID,
FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
            Assertion assertion = builder.getAssertion();

            //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
cause the signature validation to fail
            HttpResponse response = m_client.post("/MDAUserSessionService/login",
null, SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
            EntityUtils.consumeQuietly(response.getEntity());
            response = m_client.get("/MDAService/search/pos?ulat=3.75&ulng=-
2.0&llat=-2.75&llng=3.0&start=2012-06-10T12:10:00&end=2013-012-25T12:30:00", null);

            //do something with the response

        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

Note again that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```
FileInputStream isCert = new
FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);

TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);
```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```
m_client = new RestServiceClient(serverScheme, serverHost,
Integer.valueOf(serverPort), serverBasePath);

```

```
m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);
```

The next section of the code deals with the creation of the SAML assertion with the user's attributes. The AssertionBuilder is a utility provided by the MDA toolkit to aid in creating the SAML. The required attributes are defined in the attribute specification. Every assertion must provide the ElectronicIdentityID, FullName, CitizenshipCode, and entitlement attributes. Note that the example below shows how to explicitly set the expiration time for the assertion. If not included, the sessions formed by each assertion are timed-out automatically.

```
String assertingPartyID = "test.client";
AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION, 10*60);  // valid
for 10 minutes
builder.addAttribute("ElectronicIdentityId", "gfipm:2.0:user:ElectronicIdentityId",
"<a href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T. User");
```

The following code shows how to set the citizenship and entitlement information. These attributes are mapped from the consumer system's internal user database.

```
Attribute attr = builder.addAttribute("CitizenshipCode",
"mise:1.4:user:CitizenshipCode", "USA");
builder.addAttribute("LawEnforcementIndicator",
"mise:1.4:user:LawEnforcementIndicator", "true");
builder.addAttribute("PrivacyProtectedIndicator",
"mise:1.4:user:PrivacyProtectedIndicator", "true");
```

As the final step in the process to create the SAML assertion, the assertion must be signed using the private key of the consumer system. This is the same private key/key store used to establish the SSL connection. Note that this signing operation explicitly includes the assertingPartyID, which should match the entity ID of the consumer system registered with the Trust Fabric.

```
builder.signUsingPkcs12(assertingPartyID,
FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
Assertion assertion = builder.getAssertion();
```

Once the assertion has been created, the HTTP request for the session and the search can be performed. Prior to the actual request, the consuming system must establish a session with the MISE with the entitlements for the requesting user. This code makes that request using the assertion that was just created. The RestClient internally stores the session cookie provided back by the MISE to use in future requests.

```
//Important to not use SAMLUtils.asPrettyXMLString(object) as it will cause the
signature validation to fail
HttpResponse response = m_client.post("/MDAUserSessionService/login", null,
SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
EntityUtils.consumeQuietly(response.getEntity());
```

Finally, once the assertion has been created and the session established, the search request can be made. The search request shown requests all Position instances in the specified bounding box, published in the specified time period. This will return an Atom feed containing summaries of all matching records in the MISE. The retrieve operation for each of the individual records in the Atom feed is discussed in the next section.

```
response = m_client.get("/MDAService/search/pos?ulat=3.75&ulng=-2.0&llat=-
2.75&llng=3.0&start=2012-06-10T12:10:00&end=2013-012-25T12:30:00", null);
```

In production, the searching response-handling, and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the search operation.

Please note that the current base URL for the MISE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

10. Interfacing with the Retrieve Service

Once a search operation has been performed, the Retrieve interface on the MISE allows a consuming system to retrieve the complete NIEM-M XML instance of any record. Alternately, the retrieve URL for any record can be accessed directly if known, bypassing the search operation entirely.

The complete interface description for search is in the Search/Retrieve Specification. See the Code Overview page for the code download and library details. The ClientTest project containing these code examples is also available on that page.

Each record published to the MISE creates a unique retrieve URL for that record. As long as that record exists in the MISE cache, it can be accessed via that URL. The following example demonstrates the retrieve operation. As with the search, retrieve requires that the consumer system pass the entitlement attributes of the user. The meanings of the attributes are discussed in detail in the Attribute Specification. These entitlement attributes are provided to the MISE via a SAML assertion. When the SAML assertion is validated, the MISE returns a session cookie to the client system, which must be provided for all subsequent retrieve operations.

```
package test;

import gov.mda.Constants;
import gov.mda.saml.AssertionBuilder;
import gov.mda.saml.SAMLUtils;
import gov.mda.trustfabric.TrustFabric;
import gov.mda.util.RestServiceClient;

import java.io.FileInputStream;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;

import org.apache.commons.io.FilenameUtils;
import org.apache.http.HttpResponse;
import org.apache.http.entity.ContentType;
import org.opensaml.saml2.core.Assertion;
import org.opensaml.saml2.core.Attribute;

public class TestRetrieveClient {

    /**
```

```
     * @param args
     */
    public static void main(String[] args) {
        RestServiceClient m_client;

        /* Strongly recommend that these be loaded from a configuration file dynamically in production code */

        String miseCert = "C:\\Users\\user\\Documents\\ca\\ca.crt"; //public certificate for the MISE
        String trustFabricUrl = "<a href="https://mise.mda.gov/miseresources/TrustFabric.xml">https://mise.mda.gov/miseresources/TrustFabric.xml</a>"; //trust fabric URL on the MISE server
        String trustFabricBackupPath = "C:\\Users\\user\\Documents\\TrustFabricBackup.xml"; //backup local file location for a cached version of the trust fabric
        String serverScheme = "https";
        String serverHost = "mise.mda.gov";
        String serverPort = "9443";
        String serverBasePath = "/services";
        String keystorePath = "C:\\Users\\user\\Documents\\server.p12"; //keystore which contains the certificate and private key for this trusted system
        String keystorePass = "password";

        try {
            FileInputStream isCert = new FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
            CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
            X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);

            TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);

            m_client = new RestServiceClient(serverScheme, serverHost,  Integer.valueOf(serverPort), serverBasePath);
            m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath), keystorePass);

            //Form the user assertion
            String assertingPartyID = "test.client";
            AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
            builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION, 10*60);  // valid for 10 minutes
            builder.addAttribute("ElectronicIdentityId", "gfipm:2.0:user:ElectronicIdentityId", "<a href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
            builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T. User");

            Attribute attr = builder.addAttribute("CitizenshipCode", "mise:1.4:user:CitizenshipCode", "USA");
            builder.addAttribute("LawEnforcementIndicator", "mise:1.4:user:LawEnforcementIndicator", "true");
            builder.addAttribute("PrivacyProtectedIndicator", "mise:1.4:user:PrivacyProtectedIndicator", "true");

            builder.signUsingPkcs12(assertingPartyID, FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
            Assertion assertion = builder.getAssertion();

            //Important to not use SAMLUtils.asPrettyXMLString(object) as it will
```

```
cause the signature validation to fail
            HttpResponse response = m_client.post("/MDAUserSessionService/login",
null, SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);

            String entityID = "https%3A%2F%2Fmise.agencyone.gov%2F";
            String uuid = "79869883848892520";
            response = m_client.get("/MDAService/retrieve/ian?entityid=" + entityID
+ "&recordid=" + uuid, null);

            //do something with the response

        } catch(Exception e) {
            e.printStackTrace();
        }
    }
}
```

Note that all the configuration parameters should not be hardcoded as strings in production code, but should be loaded dynamically from a configuration file or configuration database.

Examining the code in detail, the following 4 lines load the SSL certificate for the MISE from the file system, and initialize the Trust Fabric. The Trust Fabric code attempts to load the MISE-hosted Trust Fabric from the MISE endpoint first, and can fall back to a local copy if the MISE endpoint cannot be accessed.

```
FileInputStream isCert = new
FileInputStream(FilenameUtils.separatorsToSystem(miseCert));
CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
X509Certificate cert = (X509Certificate) certFactory.generateCertificate(isCert);

TrustFabric.initializeFromURL(trustFabricUrl, trustFabricBackupPath, cert);
```

The next section of the code creates the MISE Rest Client and initializes it with the client key store. The client key store must contain the private key corresponding to the certificate registered for the publishing system in the Trust Fabric.

```
m_client = new RestServiceClient(serverScheme, serverHost,
Integer.valueOf(serverPort), serverBasePath);

m_client.setClientCert(FilenameUtils.separatorsToSystem(keystorePath),
keystorePass);
```

The next section of the code deals with the creation of the SAML assertion with the user's attributes. The AssertionBuilder is a utility provided by the MDA toolkit to aid in creating the SAML. The required attributes are defined in the attribute specification. Every assertion must provide the ElectronicIdentityID, FullName, CitizenshipCode, and entitlement attributes. Note that the example below shows how to explicitly set the expiration time for the assertion. If not included, the sessions formed by each assertion are timed-out automatically.

```
String assertingPartyID = "test.client";
AssertionBuilder builder = new AssertionBuilder(assertingPartyID);
builder.addStandardConditions(Constants.MISE_AUDIENCE_RESTRICTION, 10*60);  // valid
for 10 minutes
builder.addAttribute("ElectronicIdentityId", "gfipm:2.0:user:ElectronicIdentityId",
"<a href="mailto:testuser@testsystem.gov">testuser@testsystem.gov</a>");
builder.addAttribute("FullName", "gfipm:2.0:user:FullName", "Test T. User");
```

The following code shows how to set the citizenship and entitlement information. These attributes are mapped from the consumer system's internal user database.

```
Attribute attr = builder.addAttribute("CitizenshipCode",
"mise:1.4:user:CitizenshipCode", "USA");
builder.addAttribute("LawEnforcementIndicator",
"mise:1.4:user:LawEnforcementIndicator", "true");
builder.addAttribute("PrivacyProtectedIndicator",
"mise:1.4:user:PrivacyProtectedIndicator", "true");
```

As the final step in the process to create the SAML assertion, the assertion must be signed using the private key of the consumer system. This is the same private key/key store used to establish the SSL connection. Note that this signing operation explicitly includes the assertingPartyID, which should match the entity ID of the consumer system registered with the Trust Fabric.

```
builder.signUsingPkcs12(assertingPartyID,
FilenameUtils.separatorsToSystem(keystorePath), keystorePass);
Assertion assertion = builder.getAssertion();
```

Once the assertion has been created, the HTTP request for the session and the retrieve can be performed. Prior to the actual request, the consuming system must establish a session with the MISE with the entitlements for the requesting user. This code makes that request using the assertion that was just created. The RestClient internally stores the session cookie provided back by the MISE to use in future requests.

```
//Important to not use SAMLUtils.asPrettyXMLString(object) as it will cause the
signature validation to fail
HttpResponse response = m_client.post("/MDAUserSessionService/login", null,
SAMLUtils.asXMLString(assertion), ContentType.APPLICATION_XML);
```

Finally, once the assertion has been created and the session established, the retrieve request can be made. Each record has an ID that is unique to the publishing system, so the entity ID/record ID pair provides a unique key for the record. The entity ID in each case is the same as the entity ID in the trust fabric.

```
String entityID = "https%3A%2F%2Fmise.agencyone.gov%2F";
String uuid = "79869883848892520";
response = m_client.get("/MDAService/retrieve/ian?entityid=" + entityID +
"&recordid=" + uuid, null);
```

In production, the response-handling and error-handling code must be more robust, to deal with the various error codes that might be returned by the MISE depending on the interaction with the security services and the outcome of the retrieve operation.

Please note that the current base URL for the MISE is /services/MDAService/, followed by publish, search, or retrieve, as described in this guide.

11. Testing on the Test Service Platform

Once the initial development work has been completed, the operation of the code can be tested against the MISE Test Platform. The following steps detail the process to test.

All information exchanged on the Test Platform must be TEST ONLY. No operational data may be exchanged on the Test Platform

1. Contact the MISE Engineering Team to obtain a test key store. The test key store provides the private key and certificate for one of the identities in the test Trust Fabric. This will allow connections with the security services on the MISE Test Platform.
2. Test the SSL handshake process with the Test Platform. Using the test key store, it should be possible to make the initial SSL connection, enabling further testing. Test the
3. Once the SSL connection has been established, test information can be published. Test, publish, and update for success and error conditions.
4. With the SSL connection, the delete operation can also be tested. Test delete for both success and error conditions.
5. If the search and retrieve services are required, the SAML assertions for user entitlement exchange should be tested. Ensure that the trusted system code can create and sign the necessary assertions to pass user entitlement information. Once a SAML assertion is correctly passed to the Test Platform, it will return a session cookie representing that user entitlement session, enabling access to the search and retrieve services. This test process should also test that the trusted system code correctly separates sessions, handles errors, logs out, and handles SAML re-assertion when a session expires.
6. Once user entitlement sessions can be established, the search service can be tested. Test that issuing queries for previously published information is successful. Test for error conditions and handling empty responses. Finally, if a query will return too much information, the MISE will issue a 413 error code. Make sure this case is handled, typically by refining and re-issuing the query.
7. The final operation to test is retrieve. Using the results of a search, check that the full NIEM-M instance document can be retrieved. Again, make sure that error conditions are correctly handled
8. For a system that does both publish and search/retrieve, a final end-to-end test can be performed by publishing and then searching for and retrieving an information product or a set of products.

Testing a publishing system requires steps 1-4, since user entitlements and session handling is not required for publish and delete. Testing a search/retrieve consumer system requires steps 1,2, and 5-8. The MISE Engineering Team is available to help with error logs and debugging code on the MISE side. The Specifications detail all of the error code and header information that might be returned by the MISE for success and failure states in the interaction.

12. Going Live on the Integration Platform

Once the Testing is complete, the trusted system development team should perform the following actions in conjunction with the MISE Engineering Team to go live on the MISE Integration Platform.

1. Provide the public certificate for the private key that will be used for all interactions with the MISE Integration Platform.
2. The MISE engineering team will place this certificate and the trusted system information in the Trust Fabric and sign the new Trust Fabric with the MISE private key as discussed in the Security Specification.
3. At an agreed-upon time, the new Trust Fabric will be loaded on the MISE Integration Platform and made available to all trusted systems. This is a hot-reload operation, which does not require that the MISE Integration Platform be taken offline.
4. The trusted system can now begin publishing to and consuming information from the MISE. The first publish/search/retrieve operations should be monitored by both the trusted system development team and the MISE Engineering Team to ensure successful operation and if any troubleshooting is required.

APPENDIX B - ATTRIBUTE SPECIFICATION

Maritime Information Sharing Environment (MISE)

Attribute Specification

Version 1.0

25 March 2013

1. Introduction

Maritime security is a national priority that depends on the ability to efficiently, effectively, and appropriately share and safeguard information among trusted maritime partners within the Global Maritime Community of Interest (GMCOI). The Maritime Information Sharing Environment (MISE) as defined in the National Maritime Architecture Plan enables secure, standardized sharing of unclassified maritime information among a wide variety of federal, state and local agencies as well as international participants. MISE employs attribute-based access control and a standardized set of security attributes for information access policy enforcement to facilitate information sharing with non-provisioned users in a dynamic environment. This security model embraces the philosophy that user accounts are most effectively managed by a user's parent organization and exploiting existing user account verification, management, and revocation processes already in place. Allowing access using an individual's existing local user identity and password improves security by eliminating the requirement to create and maintain yet another user identity. The federated model also eliminates the need for changes to user account privileges across multiple systems; an individual need only advise their parent organization of changes in employment status or organizational role, and changes to account privileges at the local level are sufficient to ensure security across the MISE domain. By federating the identity management to the trusted systems, the cost and burden of managing user accounts is greatly reduced for the MISE thus reducing overall sustainment costs. A common set of security attributes for entities, users, and data is necessary to consistently share and protect information across the federation of systems in the MISE.

1.1. PURPOSE

This specification defines a common set of attributes used within the Maritime Information Sharing Environment (MISE) to communicate information about users and the trusted systems that connect to the Information Sharing Infrastructure (ISI) on the user's behalf.

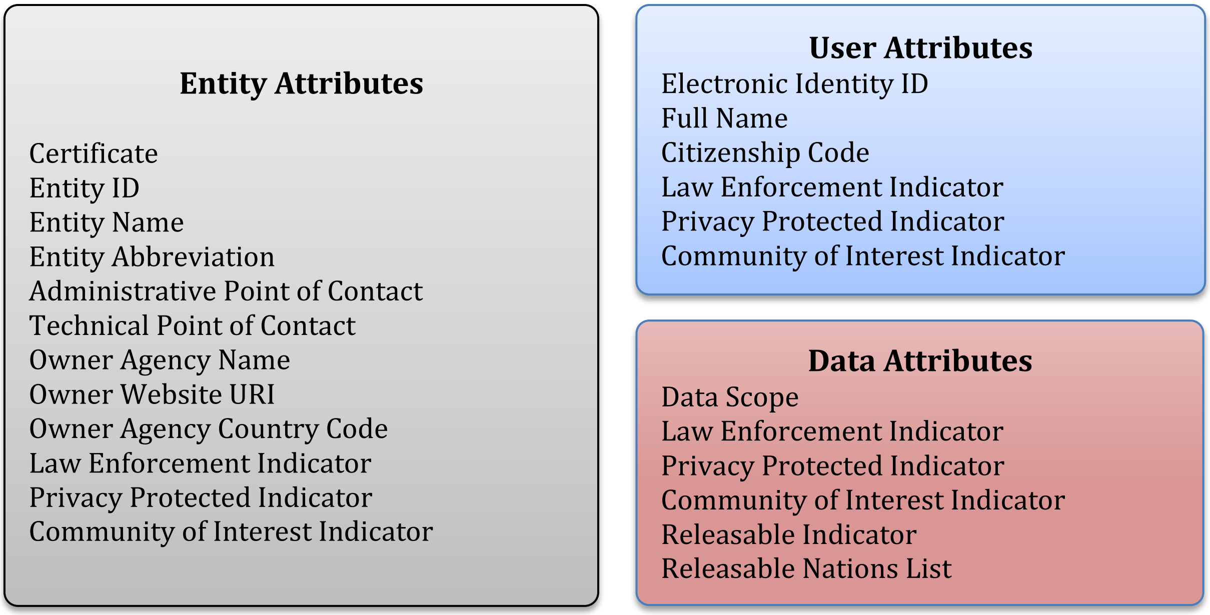
There are three categories for attributes defined for the National MDA Architecture:

1. Entity Attributes: Attributes that pertain to a trusted system within the MISE.

12. User Attributes: Attributes that pertain to a human user.

13. Data Attributes: Attributes that pertain to data.

Figure 1 summarizes the entity, user, and data attributes defined for the MISE.



Entity Attributes

Certificate
Entity ID
Entity Name
Entity Abbreviation
Administrative Point of Contact
Technical Point of Contact
Owner Agency Name
Owner Website URI
Owner Agency Country Code
Law Enforcement Indicator
Privacy Protected Indicator
Community of Interest Indicator

User Attributes

Electronic Identity ID
Full Name
Citizenship Code
Law Enforcement Indicator
Privacy Protected Indicator
Community of Interest Indicator

Data Attributes

Data Scope
Law Enforcement Indicator
Privacy Protected Indicator
Community of Interest Indicator
Releasable Indicator
Releasable Nations List

Figure 23 - MISE Attributes Summary

Entity attributes are used to capture relevant information about the trusted system i.e. SSL certificate, administrative information, and website Uniform Resource Identifier (URI). The Owner Agency Country Code is used in protecting data based on nation. An entity is restricted from accessing information unless it is explicitly coded for release to their country code. The "Indicator" attributes are key to establishing what categories of information the trusted system can access.

User attributes provide detailed information on the individual users of a trusted system, and serve to ensure that they are only able to access that subset of the available information to which they individually require and qualify for access. The user attributes are assigned and managed by the trusted system to which the user belongs.

Data attributes are fundamental to entitlement management and data management processes within the MISE. The three data attributes defined in this specification as “indicators” support entitlement management within the MISE. The “scope” attribute provides the flexibility to associate data with a specific incident or operation to provide context for data management.

1.2. INDICATORS

Entitlement management within the MISE involves run-time authorization decisions about whether a trusted system and requesting user are authorized to access a requested information resource. Three primary security indicators are used by data providers to declare information access policy to set access restrictions on information they share with MISE. These information access policies are the basis for entitlement decisions within the MISE. A fourth indicator, the “Releasable Indicator”, is a Boolean used in conjunction with the three primary Security Indicators to mark data as releasable to the public domain under the restrictions of the associated indicator. The three primary security indicators are listed in the following table in order of decreasing degree of restriction.

Indicator	Abbreviation
Law Enforcement	LEI
Privacy Protected	PPI
Community of Interest	COI

Table 1 – Security Indicators

The following describes the three primary attributes and explains their intent as well as how the attribute might be used:

Law Enforcement Indicator: This Indicator is the most restrictive, and is used to code data for release to entities such as federal, state and local law enforcement agencies. Only Entities assigned the Law Enforcement Indicator and the U.S. owner agency code can access this data. Within an Entity with the Law Enforcement Indicator, only Users who are U.S. Citizens and assigned the Law Enforcement Indicator by that Entity will be able to access the information.

Privacy Protected Indicator: Personally Identifiable Information (PII) is information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual. State and Federal legislation, as well as the policy of many agencies such as DOD and DHS, impose strict limitations on the release of PII. The Privacy Protected Indicator is designed to restrict access to and distribution of PII across the MISE Domain. Application of the Privacy Protected Indicator means the information will only be released to Entities with the requisite country code, and then only to Individuals with that attribute and the requisite country code assigned to their account.

Community of Interest Indicator: The community of interest indicator is the minimum access level assigned to trusted systems participating in the environment. By default, all trusted systems are granted the Community of Interest Indicator.

The following two modifiers can be used in conjunction with any of the three primary security indicators:

Releasable Indicator (default False): This attribute is a Boolean value to mark data as releasable to the public domain under the restrictions of the associated indicator.   This attribute is used to indicate data can be released in accordance with the consuming systems policies, business processes and as required to support their mission.

Releasable Nations List (default USA): This provides a comma-separated list of nations who can access the associated data. This is a space-separated list of three-letter country codes, for example (CAN USA FRA). This attribute is used to indicate data can be released to only those nations identified by country codes.

Scope provides tags to associate data with a specific incident or operation. These tags are used by data providers to relax or override their IAP for normal operations so trusted systems and users are able to access data they would normally be restricted from, but only within the scope of their involvement in the specified event or operation. Scope may be a planned event such as the Olympics or specific response operation in the wake of a specific disaster such as Hurricane

Katrina. Each scope is named, and provides three modifiers to the data indicators discussed in the previous section:

Scope: Name of the scope. This will indicate which event or operation within which the scope is in context, e.g. SuperstormSandy.

ScopeDataIndicator: Minimum indicator required for access within the context of this scope. If this data is normally PPI data, the data provider might want to provide it to all COI consumers within the context of this scope.

ScopeReleasable: This attribute is a Boolean value to mark data as releasable to the public domain under the restrictions of the associated indicator.   This attribute is used to indicate data is can be released in accordance with the consuming systems policies, business processes and as required to support their mission, within the context of the associated scope.

ScopeReleaseableNations: This provides a comma-separated list of nations who can access the associated data within the context of the associated scope. This is a space-separated list of three-letter country codes, for example (CAN USA FRA).

As an example, the following table shows in the case of a trusted system providing position reports, the policy during routine operations is PPI required for access because the data contains US Persons information. However, in support of disaster relief operations during Hurricane Sandy the data provider may decide they have a need to share with any trusted system within the context of Humanitarian Aid and Disaster Relief (HADR) operations so COI can be set as minimum requirement for access. The data provider can modify the indicator, releasability, and releasable nations in the context of the Hurricane Sandy scope only.

Use Case (Tracks)	Routine Operations	Indicator/Scope
HADR (US Persons)	PPI	COI / SuperstormSandy

Table 2 – Scope Example

The four attributes described for scope are not new attributes, but simply modifiers on the existing data attributes defined in section 4 of this specification.

2. Entity Attributes

The following entity attributes are associated with a trusted system.

2.1 Administrative Point of Contact – Email Address Text	
Name	AdministrativePointofContactEmailAddressText
Description	The electronic mailing address by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.

Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactEmailAddressText
Example Values	"john.doe@company.com"
2.2 Administrative Point of Contact – Fax Number	
Name	AdministrativePointofContactFaxNumber
Description	The telephone number for a facsimile device by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactFaxNumber
Example Values	"(555) 555-5555"
2.3 Administrative Point of Contact – Full Name	
Name	AdministrativePointofContactFullName
Description	The complete name of the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactFullName
Example Values	"John Doe"
2.4 Administrative Point of Contact – Telephone Number	
Name	AdministrativePointofContactTelephoneNumber
Description	The telephone number for a telecommunications device by which to contact the person who is the administrative point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:AdministrativePointofContactTelephoneNumber
Example Values	"(555) 555-5555"

2.5 Certificate	
Name	Certificate
Description	An electronic certificate used by the entity as a cryptographic trust anchor within a federation for the purposes of digital signatures and/or encryption. The certificate is represented in X.509 v3, base-64 encoded format.
Data Type	Base-64 Binary
Typical Usage	Authentication, Registration, Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:Certificate
Example Values	"MIICJzCCAZCgAwIBAgIBGDANBgkqhkiG9w0BAQUFADBAMQswCQYDVQQGEwJVUzEMMAoGA1UEChMDSVNDMSMwIQYDVQQDExpJU0MgQ0RLIFNhbXBsZSBDZXJ0aWZpY2F0ZTAeFw0wMzA3MTcwMDAwMDBaFw0wNDA3MTcwMDAwMDBaMEAxCzAJBgNVBAYTAlVTMQwwCgYDVQQKEwNJU0MxIzAhBgNVBAMTGklTQyBDREsgU2FtcGxlIENlcnRpZmljYXRlMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC9GQTkukn+153rATR8dh2Hm8ixF7f7Y7bI0VFJnJAQCKqta4/IhFwQIK5F2Gn8j9tITBiXCF7F6XSvaF8bivN10zR0pvI11NflEm2kwh7Yw0jZJB17Y3FHg183qYegmm/UwqX5zKUa4xw+cE8XSEqUuwjg0roBMGhAMzFEihHzLwIDAQABozEwLzAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAwIAYDAPBgNVHQ4ECHJzYS0xMDI0MA0GCSqGSIb3DQEBBQUAA4GBALWGxxo55ScpLfECnqEUixFwrzftQGD2ISda7EWp/d7k23fOXgHC7Za18OpvlBUZ3sC2Fg4finfRHd2J4mXONk5OEdjhJILd58GErcCECg4J2uJPz77/zk+giiXldQEPtG+YOaAbZC2SFbdfyYDKiSPhgzdy0/b4cElf4+VzegRM"
2.6 Community of Interest Indicator	
Name	COIIndicator
Description	True if the entity is allowed access to COI data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:COIIndicator
Example Values	"True," "False"
2.7 Entity Abbreviation	
Name	EntityAbbreviation
Description	An abbreviation or acronym for the name of a trusted system.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:EntityAbbreviation

Example Values	“MAGNET,” “MSSIS”
2.8 Entity ID	
Name	EntityID
Description	The unique identifier by which the entity is denoted.
Data Type	Text
Typical Usage	Registration, Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:EntityId
Example Values	“MSSIS:123,” ”MISE:TIB:MAGNET”
2.9 Entity Name	
Name	EntityName
Description	The name of an entity.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:EntityName
Example Values	“Maritime Analysis Global Network,” “Maritime Safety and Security Information System”
2.10 Law Enforcement Indicator	
Name	LawEnforcementIndicator
Description	True if the entity is allowed access to law enforcement protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:LawEnforcementIndicator
Example Values	“True,” “False”
2.11 Owner Agency Country Code	
Name	OwnerAgencyCountryCode
Description	The country of the organization or agency by which the entity is owned and operated.
Data Type	Country Code ISO 3166-1
Typical Usage	Registration
References	

Formal Name	mise:1.4:entity:OwnerAgencyCountryCode
Example Values	”USA,” “GBR,” “FRA”
2.12 Owner Agency Name	
Name	OwnerAgencyName
Description	The name of the organization or agency by which the trusted system is owned.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:OwnerAgencyName
Example Values	”NORAD-USNORTHCOM”
2.13 Owner Agency Website URI	
Name	OwnerAgencyWebSiteURI
Description	The Internet address or website of the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:OwnerAgencyWebSiteURI
Example Values	“http://website.company.com”
2.14 Privacy Protected Indicator	
Name	PrivacyProtectedIndicator
Description	True if the entity is allowed access to privacy-protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:entity:PrivacyProtectedIndicator
Example Values	“True,” “False”
2.15 Technical Point of Contact – Email Address Text	
Name	TechnicalPointofContactEmailAddressText
Description	The electronic mailing address by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration

Table 3 – List of Entity Attributes

References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactEmailAddressText
Example Values	"john.doe@company.com"
2.16 Technical Point of Contact – Fax Number	
Name	TechnicalPointofContactFaxNumber
Description	The telephone number for a facsimile device by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactFaxNumber
Example Values	"(555) 555-5555"
2.17 Technical Point of Contact – Full Name	
Name	TechnicalPointofContactFullName
Description	The complete name of the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactFullName
Example Values	"John Doe"
2.17 Technical Point of Contact – Telephone Number	
Name	TechnicalPointofContactTelephoneNumber
Description	The telephone number for a telecommunication device by which to contact the person who is the technical or engineering point of contact for the entity within the organization or agency by which the entity is owned and operated.
Data Type	Text
Typical Usage	Registration
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:entity:TechnicalPointofContactTelephoneNumber
Example Values	"(555) 555-5555"

3. User Attributes

The following user attributes are associated with an end user.

3.1 Citizenship Code	
Name	CitizenshipCode
Description	The country that has assigned rights, duties, and privileges to the user because of the birth or naturalization of the user in that country.
Data Type	ISO 3166-1 Alpha-3 Country Code
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:CitizenshipCode
Example Values	”USA,” “GBR,” “FRA”
3.2 Community of Interest Indicator	
Name	COIIndicator
Description	True if the user is allowed access to COI data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:COIIndicator
Example Values	“True,” “False”
3.3 Electronic Identity Id	
Name	ElectronicIdentityId
Description	The unique identifier that is associated with the electronic identity for the user within the user's identity provider's user base.
Data Type	Text
Typical Usage	Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:user:ElectronicIdentityId
Example Values	“DOE.JOHN.A.2370295257”
3.4 Full Name	
Name	FullName
Description	The complete name of the user.
Data Type	Text

Typical Usage	Audit Logging
References	GFIPM 2.0 Metadata Specification
Formal Name	gfipm:2.0:user:FullName
Example Values	"John Doe," "Jim Q. Public"
3.5 Law Enforcement Indicator	
Name	LawEnforcementIndicator
Description	True if the user is allowed access to law enforcement protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:LawEnforcementIndicator
Example Values	"True," "False"
3.6 Privacy Protected Indicator	
Name	PrivacyProtectedIndicator
Description	True if the user is allowed access to privacy-protected data, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:user:PrivacyProtectedIndicator
Example Values	"True," "False"

Table 4 – List of User Attributes

4. Data Attributes

The following data attributes are associated with information exchanged within the MISE.

4.1 Community of Interest Indicator	
Name	COIIndicator
Description	True if the data is COI protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:CommunityOfInterestIndicator
Example Values	"True," "False"
4.2 Data Scope	
Name	Scope
Description	An indicator which denotes a scope, situation, or event of interest for which data is deemed relevant and/or accessible.
Data Type	Text
Typical Usage	Authorization
References	
Formal Name	mise:1.4:data:Scope
Example Values	"HurricaneKatrina," "Baltimore1812Exercise," "OperationTomadachi"
4.3 Law Enforcement Indicator	
Name	LawEnforcementIndicator
Description	True if the data is law enforcement-protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:LawEnforcementIndicator
Example Values	"True," "False"
4.4 Privacy Protected Indicator	
Name	PrivacyProtectedIndicator
Description	True if the data is privacy-protected, false otherwise.
Data Type	Boolean
Typical Usage	Authorization, Audit Logging

References	
Formal Name	mise:1.4:data:PrivacyProtectedIndicator
Example Values	“True,” “False”
4.5 Releasable Indicator	
Name	Releasable Indicator
Description	True if the data is publicly releasable
Data Type	Boolean
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:ReleasableIndicator
Example Values	“True,” “False”
4.6 Releasable Nations	
Name	Releasable Nations Code List
Description	A space separated list of countries with assigned rights, duties, and privileges to access the data.
Data Type	ISO 3166-1 Alpha-3 Country Code
Typical Usage	Authorization, Audit Logging
References	
Formal Name	mise:1.4:data:ReleasableNationsCodeList
Example Values	”USA GBR FRA”

Table 5 – List of Data Attributes

APPENDIX C - INTERFACE SECURITY SPECIFICATION

National MDA Architecture

Interface Security Specification

Version 1.0

25 March 2013

1. Introduction

The Maritime Information Share Environment (MISE) specifications for services such as Publication, Search, and Retrieval define Representational State Transfer (RESTful) interfaces specific to each service, but do not discuss specifics regarding how those interfaces are secured, how identity of the trusted system invoking the service is guaranteed, or how authenticated user attributes are delivered when applicable. The purpose of this specification is to discuss these aspects, which apply to all MISE services, and augment each RESTful interface defined in separate specifications. In addition, sample code demonstrating many aspects of MISE interface security is available at https://mise.mda.gov.

The *Security Architecture View* section of the *National Maritime Architecture Framework* should be reviewed prior to reading this specification, as it provides important overview and context to the material presented here. This specification does not repeat that overview and context information, but assumes the reader is familiar with it.

The design of MISE interface security has followed a number of patterns used in the U.S. Department of Justice's *Global Federated Identity and Privilege Management* (GFIPM). There are some fundamental architectural differences between GFIPM and the MISE, which impact the design. Key among these are:

- MISE uses a hub and spoke network topology whereas GFIPM uses a point-to-point topology.
- MISE uses RESTful service interfaces whereas GFIPM uses Simple Object Access Protocol (SOAP)-based interfaces.

Despite these differences, there is a strong relationship between MISE and GFIPM. Portions of GFIPM documents are therefore incorporated into this document and modified to fit the MISE. In particular, [GFIPM Trust] and [GFIPM Services] were referenced in creation of this specification.

1.1. REFERENCES

RFC 2119	"RFC 2119—Key Words for Use in RFCs to Indicate Requirement Levels" Internet RFC/STD/FYI/BCP Archives http://www.ietf.org/rfc/rfc2119.txt
GFIPM Trust	Federated Identity and Privilege Management (GFIPM): Cryptographic Trust Model http://www.it.ojp.gov/gist/Document/73
GFIPM Services	Federated Identity and Privilege Management (GFIPM): Web Services System-to-System Profile http://www.it.ojp.gov/gist/Document/122
SAML2 Metadata	"Metadata for the OASIS Security Markup Language (SAML) V2.0" OASIS Standard, 15 March 2005 Document Identifier: saml-metadata-2.0-os http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

SAML2 Core	"Assertions and Protocol for the OASIS Security Markup Language (SAML) V2.0" OASIS Standard, 15 March 2005 Document Identifier: saml-core-2.0-os http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
NIST SP 800-52	Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations National Institute of Science and Technology (NIST) Special Publication 800-52 http://csrc.nist.gov/publications/nistpubs/800-52/SP800-52.pdf

1.2. DOCUMENT STRUCTURE

Section 2 below describes details regarding key aspects of MISE interface security introduced in the *Security Architecture View* section of the *National Maritime Architecture Framework*.

Section **Error! Reference source not found.** presents details of the Trust Fabric document.

Section 4 presents details of the Security Assertions Markup Language (SAML) assertions.

2. Process Flow and Processing Rules

This section discusses details regarding key aspects of MISE interface security, including the purpose of each as well as important implementation and operational requirements. Figure 24 below illustrates these key aspects, and will be referenced throughout this section.

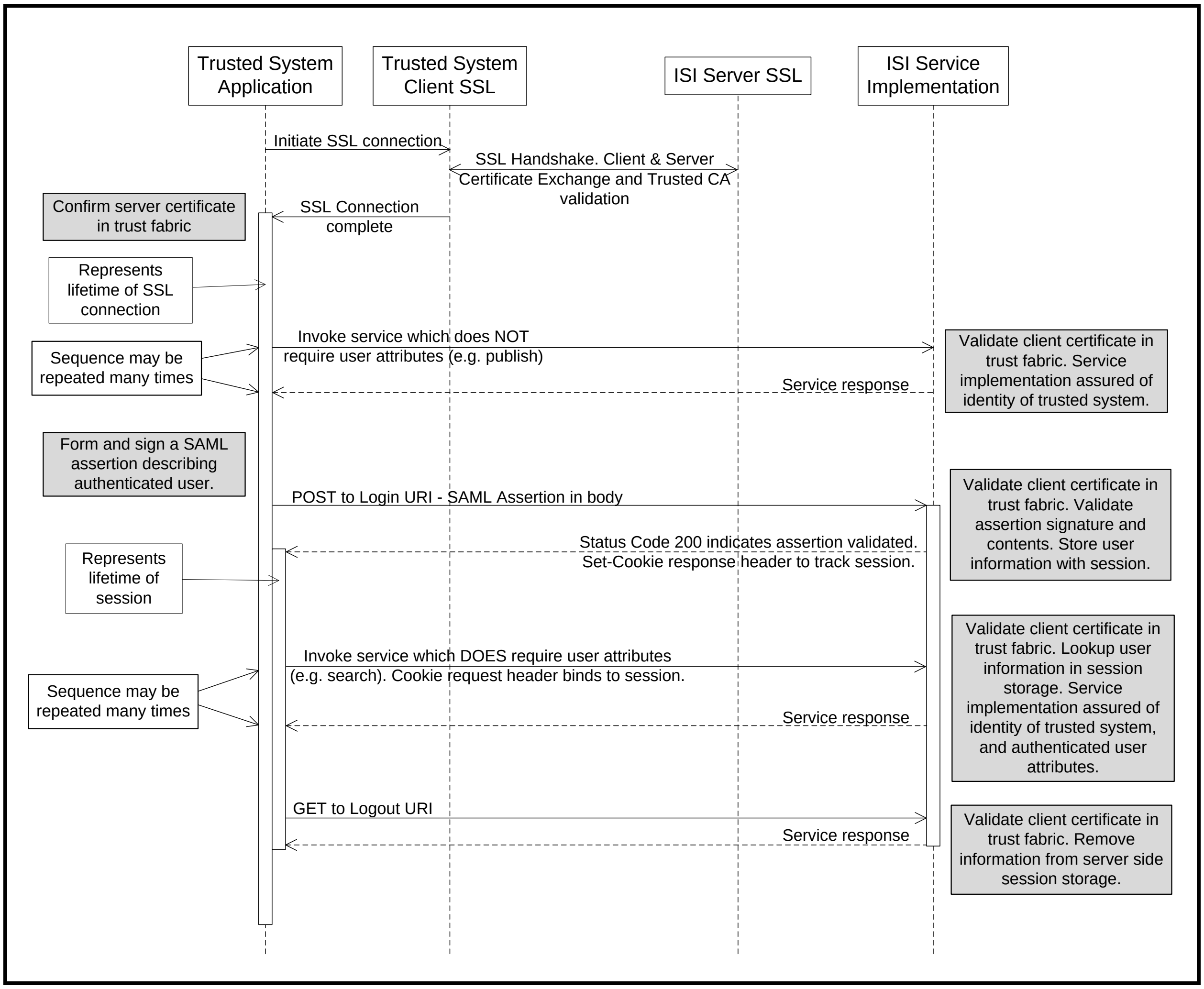

Trusted System Application
Trusted System Client SSL
ISI Server SSL
ISI Service Implementation
Initiate SSL connection
SSL Handshake. Client & Server Certificate Exchange and Trusted CA validation
Confirm server certificate in trust fabric
SSL Connection complete
Represents lifetime of SSL connection
Invoke service which does NOT require user attributes (e.g. publish)
Sequence may be repeated many times
Validate client certificate in trust fabric. Service implementation assured of identity of trusted system.
Service response
Form and sign a SAML assertion describing authenticated user.
POST to Login URI - SAML Assertion in body
Validate client certificate in trust fabric. Validate assertion signature and contents. Store user information with session.
Status Code 200 indicates assertion validated. Set-Cookie response header to track session.
Represents lifetime of session
Validate client certificate in trust fabric. Lookup user information in session storage. Service implementation assured of identity of trusted system, and authenticated user attributes.
Invoke service which DOES require user attributes (e.g. search). Cookie request header binds to session.
Sequence may be repeated many times
Service response
GET to Logout URI
Validate client certificate in trust fabric. Remove information from server side session storage.
Service response

Figure 24 - Service Invocation Process Flow

2.1. X.509 CERTIFICATES AND PRIVATE KEYS

X.509 certificates and associated private keys are used for several purposes in MISE interface security. Specifically:

- Signing the trust fabric document by the MISE Certificate Authority (CA).
- Signing Security Assertions Markup Language (SAML) assertions (which contain user attributes) by information consumer systems.

- Client and server certificates for securing Secure Sockets Layer (SSL) connections between trusted systems and the Information Sharing Infrastructure (ISI).

2.1.1. PUBLIC / PRIVATE KEY PAIR

The process of creating an X.509 certificate begins with generating a pair of keys that are mathematically related - a *public key* and a *private key*. One key locks or encrypts data, and the other unlocks or decrypts the data. Neither key can perform both functions by itself.[9] The private key, as its name implies, must be kept secure. The public key is included inside the X.509 certificate, and must be available to any entity needing to engage in secure interaction with the possessor of the private key.

To be used in the MISE, all generated key pairs must be **2048-bit RSA[10] keys**.

More detail regarding each usage of X.509 certificates in MISE interface security is provided in subsequent sections of this document. For each usage, Table 4 shows which system has access to the private key, and how the corresponding X.509 certificate is distributed. The rightmost column (certificate signing) is discussed in the next section.

Use	Private Key	X.509 Certificate Distribution	X.509 Certificate Signing
Signing trust fabric document	MISE Management	Provided to each trusted system during on-boarding process.	MISE CA
Signing SAML assertions	Information consumer system which asserts user attributes	Included in trust fabric	Any well-known root CA
ISI SSL server certificate	ISI	Included in trust fabric	Any well-known root CA
Trusted system SSL client certificate	Trusted system	Included in trust fabric	Any well-known root CA

Table 4 - MISE X.509 Certificate Uses

2.1.2. MISE CERTIFICATE AUTHORITY (CA)

The MISE Management operates a certificate authority (CA) to provide trust and security to the environment. The sole purpose of this CA is to sign the Trust Fabric. The CA does not issue certificates to trusted systems. Trust in the MISE is anchored by the Trust Fabric document and the MISE CA's signature of the document.

[9] See http://en.wikipedia.org/wiki/Public-key_cryptography for more background information.

[10] RSA stands for Rivest, Shamir, Adelman; the surnames of the computer scientist and two cryptographers who developed the algorithm.

2.1.3. X.509 CERTIFICATES

Before a keypair can be used in the MISE for secure interaction, an X.509 certificate must be created containing the public key. An X.509 certificate binds a name to a public key value. The role of the certificate is to associate a public key with the identity contained in the X.509 certificate. All certificates must be digitally signed by a CA. A CA is a trusted entity that confirms the integrity of the public key value in a certificate. To be used in the MISE, an X.509 certificate must be signed by a well-known *root certificate authority*[11]. Any root CA trusted by all major browsers is acceptable.

CREATING X.509 CERTIFICATES

Numerous tools and processes are available for creating key pairs and X.509 certificates. The exact process chosen by a trusted system will vary depending on the platform the trusted system implementation is based upon, agency procedures, and the chosen root CA.

In some cases a trusted system will choose to generate a keypair and a certificate signing request (CSR) internally using a tool such as OpenSSL[12] or Java's keytool, and submit the CSR to the root CA for signing. In other cases, a trusted system may choose to use tools provided by the root CA for generation of the keypair in addition to signing the certificate. For step-by-step instructions, see the Implementation Guide at https://mise.mda.gov.

2.2. SSL CONNECTIONS

All MISE service invocations must take place over SSL network connections. Both server and client SSL certificates are required. The top portion of Figure 24 illustrates establishing an SSL connection.

The following requirements are standard with SSL connections, and are part of any SSL implementation:

- The client side must validate the signature of the certificate presented by the server, confirm that it was signed by a root CA trusted by the client, and confirm that the server proves possession of the private key associated with the certificate.
- The client side must validate that the Subject common name (CN) within the server certificate matches the domain name portion of the service Uniform Resource Locator (URL) endpoint.
- The server side must validate the signature of the certificate presented by the client, confirm that it was signed by a root CA trusted by the server, and confirm the client proves possession of the private key associated with the certificate.

Beyond these standard SSL requirements, the MISE requires:

- Each side of the SSL connection must confirm that the certificate presented by the opposite side exists within a **`RoleDescriptor`** element of the Trust Fabric, and must

[11] See http://en.wikipedia.org/wiki/Root_certificate for more background information.

[12] http://www.openssl.org/

have **`used="signing"`** on the **`KeyDescriptor`**. (Details of Trust Fabric document format are presented in section **Error! Reference source not found.**.)

- SSL v3 or Transport Layer Security (TLS) 1.1 (and higher) must be used. TLS 1.2 is recommended. In addition, it is recommended that the TLS implementation conform to [NIST SP 800-52].

Taken together, these requirements guarantee a secure point-to-point communication channel between the client (trusted system) and server (ISI). In addition each side knows the identity of the other side, knows the other side is a current approved member of the MISE, and has all metadata in the trust fabric about the other side available to it.

SSL connections are typically reused for a series of service invocations (Figure 24 illustrates this reuse). There is no requirement that a related series of service invocations take place over the same SSL connection; however, there is a requirement that the verification steps enumerated above be confirmed each time a new connection is established.

2.3. SAML Assertion Processing

As illustrated in the lower portion of Figure 24, some MISE service invocations (e.g. Search, Retrieval) are done on behalf of an individual user, or a set of users with identical user attributes. In these cases, authenticated user attributes must be available to the service.

When user attributes are required, they are delivered using SAML assertions. In SAML terms, information consumer systems act as *identity providers*. Prior to invoking services on behalf of users, an information consumer system creates a SAML assertion in accordance with the guidelines specified in section 4.1 below. This SAML assertion contains authenticated *user attributes* pertaining to the user or group of users who will be granted access to the information obtained from the subsequent series of service invocations. The assertion is then digitally signed by the information consumer system using the private key associated with the signing certificate, which is a part of its **MISEConsumerDescriptor** (see section **Error! Reference ource not found.**) role information within the trust fabric document. This provides a cryptographic guarantee to the ISI, and to information provider systems, of the identity of the information consumer system asserting the user attributes and promising to handle the information in accordance with MISE guidelines.

Rather than sending the SAML assertion with each service invocation, assertions are sent once at the beginning of a series of service invocations on behalf of a user or group of users. There is significant overhead associated with delivering the SAML assertion, cryptographically validating the digital signature, and validating the contents of the assertion document. Sending the assertion once to apply to a series of service invocations reduces the overall impact of this overhead. In addition, this pattern allows service interfaces to be fully RESTful. Request and response message bodies simply contain information associated with the service, without each interface needing to accommodate inclusion of a SAML assertion document.

Figure 24 above illustrates the pattern used to allow a single SAML assertion to be bound to multiple service invocations. Key points related to this pattern include:

- Some MISE services require user attributes, which are delivered in signed SAML assertions. Other services do not require user attributes. Whether or not a specific service requires user attributes is specified in the individual service documentation. If a service that requires user attributes is invoked without associating a SAML assertion, the service will return an HTTP status code of 403 (Forbidden) and a MISE error code of 104 (SAML assertion required but missing)(see section 2.4.1).

- All service invocations, including the Login and Logout services, must occur over SSL connections validated against the trust fabric, with the identity of the invoking trusted system guaranteed as described in section 2.2 above.
- Before invoking a service that requires user attributes, an information consumer system must create a SAML assertion and send it to the ISI using the Login service specified in section 2.3.1 below. The login service performs full validation of the assertion signature and contents, and creates a session context for subsequent service invocations to reference. The session key is returned to the information consumer system in a **`Set-Cookie`** HTTP response header.
- The information consumer system may then proceed with any number of MISE service invocations on behalf of the user or group of users as asserted with the SAML assertion. Each of these service invocations MUST include a **`Cookie`** HTTP request header whose value is the session key set by the Login request. The information consumer system has the responsibility to ensure that information retrieved from the ISI will only be made available to users described in the SAML assertion.
- Sessions and SAML assertions both have a defined lifetime. If a client sends a session key referencing a session which has expired, the MISE interface security implementation will not deliver user attributes to the service implementation; which means the error response the trusted system will see is the same as if a session key had not been sent. Specifically this will be an HTTP status code of 403 (Forbidden) and a MISE error code of 104 (SAML assertion required but missing)(see section 2.4.1). The client trusted system should respond to this error condition by invoking the login service again with a valid SAML assertion, then re-invoking the failed service request with the new session cookie returned by the login service.
- Sessions expire after 20 minutes of inactivity. Any service invocation referencing the session resets this 20-minute timer. In addition, it is recommended that trusted systems invoke the Logout service (see section 0) if the trusted system is able to determine that the session will no longer be used. This allows the ISI to free resources used to store the session information prior to the 20-minute automatic expiration period.
- SAML assertions expire when outside the time window expressed in the **NotBefore** and **NotOnOrAfter** attributes of the **`<Conditions>`** element. MISE sessions automatically expire when outside this time window, separately from the 20-minute inactivity timer.
- An information consumer system will commonly serve many simultaneous users, and thus will commonly have many simultaneous open sessions with the ISI. It is the responsibility of the information consumer system to ensure that the correct session key is used when invoking services on behalf of a given user or group of users.
- Figure 24 shows a series of service invocations operating in the context of a single SSL connection for simplicity of illustration; however this is not a requirement. Information consumer systems should keep SSL connection(s) to the ISI open during active communication to enhance performance by avoiding the overhead of establishing a fresh

SSL connection for each service invocation. However, there is no requirement that all service invocations for a given session occur over the same SSL connection. In addition, service invocations for multiple sessions may take place over a single SSL connection. It IS REQUIRED that full SSL certificate validation back to the trust fabric take place every time a fresh SSL connection is established, so if a connection pooling system is used, trust fabric validation must be integrated with the connection pool.

2.3.1. LOGIN SERVICE INTERFACE

URI	From MISELoginService element in MISEInfrastructureDescriptor role in trust fabric.	Example: https://mda.gov/services/login
Method	POST	
Request Headers		None specified
Request Content Type	application/xml	
Request Content	Signed SAML Assertion	
Status Codes	200 (OK)	Successful. Response content as described below.
	Other values	If error conditions are encountered, the response will be in accordance with section 0 below.
Response Headers	Set-Cookie: MISESession=xxxxxxxx; Path=/	"xxxxxxxx" will be replaced by a randomly-generated value unique to the session.
Response Content		Empty

Table 2 – Login Service Interface

2.3.2. LOGOUT SERVICE INTERFACE

URI	From MISELogoutService element in MISEInfrastructureDescriptor role in trust fabric.	Example: https://mda.gov/services/logout
Method	GET	
Request Headers	Cookie: MISESession=xxxxxxxx	"xxxxxxxx" is the key of the session to terminate, previously returned from Login service.
Request Content		Empty
Status Codes	200 (OK)	Successful. Response content as described below.
	Other values	If error conditions are encountered, the response will be in accordance with section 0 below.

Response Headers	None specified
Response Content	Empty

Table 3 – Logout Service Interface

2.4. MISE Error Response Content

MISE service invocations which result in HTTP status codes in the 4xx (Client Error) and 5xx (Server Error) ranges may return an XML document as the response content providing additional details about the error encountered. The HTTP status is the important code to determine the server response, and the response content is provided only for debugging purposes. When such an error response document is returned, the response **`Content-Type`** is **`application/xml`**, and the response document is in the following format:

```
<MISEError>
   <Code>100</Code>
   <Description>Client certificate not presented during SSL handshake</Description>
</MISEError>
```

2.4.1. MISE Error Codes for Interface Security

Table 4 lists MISE error codes relevant to interface security, which pertain to all MISE services. Additional service-specific error codes may be defined in individual service specification documents.

MISE Error Code	HTTP Status Code Returned	Error Description
100	403	Client certificate not presented during SSL handshake
101	500	Internal server error accessing trust fabric
102	403	Client certificate not found in trust fabric
103	403	Session cookie not associated with trusted system
104	403	SAML assertion required but missing
201	400	SAML assertion signature validation failed
202	403	SAML signing certificate not in trust fabric
203	403	SAML signing certificate not associated with trusted system
204	400	SAML assertion issued by different entity than sender
205	400	MISE SAML assertions MUST NOT include a Subject
206	400	MISE SAML assertions MUST NOT include AuthnStatement
207	400	MISE SAML assertions MUST include Conditions element
208	400	NotBefore condition of assertion failed
209	400	NotOnOrAfter condition of assertion failed
210	400	MISE SAML assertions MUST include single AudienceRestriction element

211	400	MISE SAML assertions MUST include AudienceRestriction of 'urn:mise:all'
212	403	User attribute ‘<formalName>’ disallowed by trust fabric
213	403	Asserting trusted system is not an information consumer system
299	500	Internal server error processing SAML assertion

Table 4 - MISE Interface Security Error Codes

2.5. Trust Fabric Lifecycle Management Procedures

This section describes policies and procedures used to manage the MISE Cryptographic Trust Fabric. It includes details about how the Trust Fabric is created and distributed, as well as the conditions under which the Trust Fabric is updated.

2.5.1. Trust Fabric Creation Procedure

Upon occurrence of a triggering condition for a Trust Fabric update (see section 2.5.3), the Trust Fabric must be regenerated. The process of generating a new Trust Fabric document consists of two basic operations: editing the document to reflect the desired policy change (e.g., new trusted system added to the environment) and digitally signing the new document with the MISE CA private key. The following steps describe the process in more detail.

1. Starting with the most recent Trust Fabric document, edit the document as needed to incorporate the necessary changes.
2. Copy the edited Trust Fabric document to removable media.
3. Connect the removable media containing the unsigned Trust Fabric document to the physical machine on which the signing operation will be performed. Also connect the removable media containing the CA private key to the machine.
4. Perform the cryptographic signing operation on the Trust Fabric document using the CA private key. At no point during this operation shall the CA private key be copied from the removable media onto any other storage device. Also, at no point during this operation shall the physical machine be connected to a network.
5. Copy the signed Trust Fabric document onto the removable media that contains the unsigned Trust Fabric document.

2.5.2. Trust Fabric Distribution Procedure

Upon the occurrence of a triggering condition for a Trust Fabric update, and after the generation and signing of a new Trust Fabric document, the new Trust Fabric document must be distributed to all trusted systems. The following steps describe the process in more detail.

1. Publish the new Trust Fabric document at a well-known URL.
2. Notify all trusted systems of the new Trust Fabric document via the technical contact points they have provided.

Note that while the integrity of the Trust Fabric document is paramount to the security of the federation, the Trust Fabric need not necessarily be kept confidential. The security of the MISE does not rely on the contents of the trust fabric document being kept secret, but upon its accuracy being guaranteed by the MISE CA signature. Therefore, it is permissible for the Trust

Fabric URL to be publicly accessible, and encryption of the Trust Fabric document is not necessary.

2.5.3. TRIGGERING CONDITIONS FOR TRUST FABRIC UPDATES

The following events shall constitute cause for a Trust Fabric regeneration and redistribution.

1. A new trusted system joins the MISE.
14. An existing trusted system leaves the MISE.
15. An existing trusted system undergoes a configuration change that affects its entry in the trust fabric (e.g., certificate expiration, migration to a new server, key compromise on a server, etc.).
16. The MISE CA public key certificate expires.
17. It is suspected that the MISE CA private key has been compromised.

2.5.4. TRUSTED SYSTEM RETRIEVAL AND USAGE OF TRUST FABRIC

A trusted system implementation MUST retrieve the trust fabric document from the well-known URL when initially connecting to the MISE, and promptly when notified of a change by the MISE Management. It is RECOMMENDED that trusted systems be implemented in a manner that allows the trust fabric to be hot-reloaded while the trusted system is operational. In addition, it is RECOMMENDED that trusted system implementations automatically periodically retrieve the current trust fabric document from the well-known URL, and activate the new version in the running system if it has changed.

The following verification steps MUST be performed by the trusted system each time the trust fabric document is parsed and loaded into the trusted system for use, to ensure the trust fabric document put into use is indeed the official version created and signed by the MISE Management:

1. The digital signature contained within the trust fabric document MUST be validated.
2. The certificate used to sign the trust fabric MUST be compared against the MISE CA certificate, which is delivered to the trusted system upon joining the MISE by a separate out-of-band process.
3. HTTPS is REQUIRED to retrieve the trust fabric from the well-known URL. A client certificate is not required. This allows the trust fabric document to be retrieved for examination and supports a wide variety of trusted system administrative procedures.
4. The trusted system SSL configuration MUST validate the common name of the server certificate presented when connecting to the well-known URL against the domain name of the URL, and confirm the server certificate presented is signed by a CA trusted in the MISE (see section 2.2).

Sample code (written in Java) is available at https://mise.mda.gov to demonstrate loading, validating, and automatic periodic hot reloading of the trust fabric from a well-known URL. Since the trust fabric document is a SAML metadata file with a few simple extensions, this sample code is able to leverage the open source OpenSAML project to simplify implementation. Trusted system implementations not written in Java, or which already include other SAML implementations, may also be able to simplify implementation by relying on existing SAML metadata implementations.

3. MISE Trust Fabric Document Format

3.1. TRUST FABRIC DOCUMENT SPECIFICATION

At a technical level, trust between all communications endpoints in the MISE is implemented using a combination of client and server TLS certificates, and the SAML 2.0 standard for federated system entity metadata. Information necessary to enforce trust is delivered to participants via the Trust Fabric document, which defines the most current cryptographic security context of the MISE. The document contains an **`<md:EntityDescriptor>`** entry for each communications endpoint in the environment, including the ISI, Information Provider Systems, and Information Consumer Systems. The MISE Management maintains the document and makes a new version of it available to trusted systems whenever the membership changes because of the addition or removal of a trusted system. To ensure compliance with the current Trust Fabric, each communications endpoint MUST incorporate the most current version of the Trust Fabric document into its security policy decisions in a timely fashion. The MISE Management will advise trusted systems of the urgency with which a new Trust Fabric document must be incorporated when the new document is made available. When the new Trust Fabric document is being published because of a security or trust violation, or because of the removal of a trusted system for disciplinary reasons, it is imperative that members incorporate the new Trust Fabric document as soon as is reasonably possible, and as a best practice not more than 24 hours after its release.

The MISE Trust Fabric document conforms to the specification defined in [SAML2 Metadata]. It also uses an extension schema for the **`<md:RoleDescriptor>`** element. This extension schema defines the three extensions to **RoleDescriptorType** listed below. These extensions are defined rather than using roles defined in SAML 2.0 specifications (such as **SPSSODescriptor** or **IDPSSODescriptor**) so that MISE roles and associated information can be stated explicitly in the trust fabric without implying characteristics of SAML service providers and identity providers which are not used in MISE.

1. **MISEInfrastructureDescriptor** - this role is only present within the **`<md:EntityDescriptor>`** entry defining the ISI.
2. **MISEConsumerDescriptor** - this role is present within the entry for any trusted system that acts as an Information Consumer System.
3. **MISEProviderDescriptor** - this role is present within the entry for any trusted system that acts as an Information Provider System.

Each **`<md:EntityDescriptor>`** must include at least one of these roles. A trusted system entry may include both the **MISEConsumerDescriptor** role and the **MISEProviderDescriptor** role.

Section 4.2 contains this extension schema, and Section 3.2 contains a sample Trust Fabric document conformant with these requirements.

3.1.1.SAML <ENTITIESDESCRIPTOR> ELEMENT REQUIREMENTS

The following additional requirements apply to the **`<EntitiesDescriptor>`** element, which is the top-level XML element within the MISE Trust Fabric document. These requirements supplement the requirements described in [SAML2 Metadata].

1. The **Name** attribute within **<EntitiesDescriptor>** MUST be present.
2. The **validUntil** attribute within **<EntitiesDescriptor>** MUST be present.
3. The **<ds:Signature>** element within **<EntitiesDescriptor>** MUST be present.
4. The **<Extensions>** element within **<EntitiesDescriptor>** MUST NOT be present.
5. Nested **<EntitiesDescriptor>** elements within the top-level **<EntitiesDescriptor>** MUST NOT be present.
6. One or more **<EntityDescriptor>** elements within **<EntitiesDescriptor>** MUST be present.

3.1.2. SAML <EntityDescriptor> Element Requirements

The following requirements apply to **<EntityDescriptor>** elements that appear in the Trust Fabric document. Each **<EntityDescriptor>** element provides entity metadata for a specific communications endpoint (ISI or trusted system). These requirements supplement the requirements described in [SAML2 Metadata].

1. The **entityID** attribute within **<EntityDescriptor>** MUST be present, and MUST be set to the value that was agreed upon for this entity between the entity and the MISE Management. (The entity (trusted system) chooses its **entityID** value, but the choice MUST be approved by the MISE Management.)
2. The **<ds:Signature>** element within **<EntityDescriptor>** MUST NOT be present.
3. The **<EntityDescriptor>** element for the ISI MUST contain exactly one **<RoleDescriptor> element of type MISEInfrastructureDescriptorType**. The **<EntityDescriptor>** element for each trusted system must contain either a **<RoleDescriptor> element of type MISEConsumerDescriptorType** or a **<RoleDescriptor> element of type MISEProviderDescriptorType**, and may contain one of each.
4. Each **<EntityDescriptor>** element MUST contain at least one **<ContactPerson>** element with each technical **contactType**. An **<EntityDescriptor>** element MAY contain additional **<ContactPerson>** elements.
5. The following requirements apply to each **<ContactPerson>** element within an **<EntityDescriptor>** element.
    a. The **<Extensions>** element MUST NOT be present.
    b. The **<Company>** element MUST be present.
    c. The **<GivenName>** element MUST be present.
    d. The **<SurName>** element MUST be present.
    e. At least one **<EmailAddress>** element MUST be present.
    f. At least one **<TelephoneNumber>** element MUST be present.

6. The **<AdditionalMetadataLocation>** element within **<EntityDescriptor>** MUST NOT be present.
7. Each **<EntityDescriptor>** element MAY contain one **<Extensions>** element, and the **<Extensions>** element MAY contain one or more **<gfipm:EntityAttribute>** elements as defined by the GFIPM Entity Attribute Extension Schema.

3.1.3. SAML ‹RoleDescriptor› Element Requirements

RoleDescriptor types defined by the MISE trust fabric extension schema are instantiated in the trust fabric document by specifying the **xsi:type** attribute on a **<RoleDescriptor>** element. For example:

```
<md:RoleDescriptor xsi:type="mise:MISEConsumerDescriptorType"
   protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
   ...
</md:RoleDescriptor>
```

Requirements for each MISE extension role are detailed in the following subsections.

MISEInfrastructureDescriptor Role Requirements

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**mise:MISEInfrastructureDescriptorType**" as its value.
2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**urn:oasis:names:tc:SAML:2.0:protocol**" as its value.
3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of "**signing**" MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.
6. The **<MISELoginService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of "**urn:mise:bindings:REST**".
7. The **<MISELogoutService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of "**urn:mise:bindings:REST**".
8. The **<MISESearchService>** element within **<RoleDescriptor>** MUST be present, and must contain a Binding attribute with a value of "**urn:mise:bindings:REST**".

MISEConsumerDescriptor Role Requirements

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**mise: MISEConsumerDescriptorType**" as its value.
2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**urn:oasis:names:tc:SAML:2.0:protocol**" as its value.

3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of "**signing**" MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.

MISEPROVIDERDESCRIPTOR ROLE REQUIREMENTS

1. The **xsi:type** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**mise: MISEProviderDescriptorType**" as its value.
2. The **protocolSupportEnumeration** attribute within **<RoleDescriptor>** MUST be present, and MUST have "**urn:oasis:names:tc:SAML:2.0:protocol**" as its value.
3. The **<ds:Signature>** element within **<RoleDescriptor>** MUST NOT be present.
4. One or more **<KeyDescriptor>** elements containing a use attribute with a value of "**signing**" MUST be present within **<RoleDescriptor>**.
5. Exactly one **<ds:KeyInfo>** element MUST be present within each **<KeyDescriptor>** element. Exactly one **<ds:X509Data>** element MUST be present within the **<ds:KeyInfo>** element. Exactly one **<ds:X509Certificate>** element MUST be present within the **<ds:X509Data>** element.

3.2. SAMPLE TRUST FABRIC DOCUMENT

The diagram below contains a sample of the Trust Fabric Document provided by the MISE Management to each trusted system during the onboarding process.

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntitiesDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
   xmlns:gfipm="http://gfipm.net/standards/metadata/2.0/entity"
   xmlns:mise="http://mda.gov/standards/trustfabric/1.0"
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   Name="Maritime Information Sharing Environment Trust Fabric"
   validUntil="2015-11-15T00:00:00.000Z"
   xsi:schemaLocation="urn:oasis:names:tc:SAML:2.0:metadata saml20/saml-schema-
metadata-2.0.xsd http://mda.gov/standards/trustfabric/1.0 mise-trust-fabric-
extension.xsd http://gfipm.net/standards/metadata/2.0/entity gfipm-entity-attribute-
2.0.xsd">
   <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:SignedInfo>
         <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
         <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
         <ds:Reference URI="">
            <ds:Transforms>
               <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
               <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
```

```
                     <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="mise xs" />
                 </ds:Transform>
              </ds:Transforms>
              <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
              <ds:DigestValue>q64PXRBBjnoTNL3Bg4ShLDSPrBw=</ds:DigestValue>
          </ds:Reference>
       </ds:SignedInfo>
       <ds:SignatureValue><!-- Base 64 encoded signature--></ds:SignatureValue>
       <ds:KeyInfo>
          <ds:X509Data>
              <ds:X509Certificate>
                 <!-- Base 64 encoded certificate embedded here
                     This is the MISE CA certificate
                     used to sign the trust fabric document.
                     -->
              </ds:X509Certificate>
          </ds:X509Data>
       </ds:KeyInfo>
   </ds:Signature>
   <md:EntityDescriptor entityID="https://isi.mda.gov/">
       <md:RoleDescriptor xsi:type="mise:MISEInfrastructureDescriptorType"
   protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
          xsi:type="mise:MISEInfrastructureDescriptorType">
          <md:KeyDescriptor use="signing">
              <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                 <ds:X509Data>
                     <ds:X509Certificate>
                        <!-- Base 64 encoded certificate embedded here
                           This is the server certificate which the ISI will present
                           during SSL connection handshake.-->
                     </ds:X509Certificate>
                 </ds:X509Data>
              </ds:KeyInfo>
          </md:KeyDescriptor>
          <md:KeyDescriptor use="signing">
              <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                 <ds:X509Data>
                     <ds:X509Certificate>
                        <!-- Base 64 encoded certificate embedded here
                           This is the MISE CA
                           certificate used to sign the trust fabric document. -->
                     </ds:X509Certificate>
                 </ds:X509Data>
              </ds:KeyInfo>
          </md:KeyDescriptor>
          <mise:MISELoginService Binding="urn:mise:bindings:REST"
              Location="https://isi.mda.gov/service/login" />
          <mise:MISELogoutService Binding="urn:mise:bindings:REST"
              Location="https://isi.mda.gov/service/logout" />
          <mise:MISESearchService Binding="urn:mise:bindings:REST"
              Location="https://isi.mda.gov/service/search" />
       </md:RoleDescriptor>
       <md:Organization>
          <md:OrganizationName xml:lang="en">MISE
          </md:OrganizationName>
          <md:OrganizationDisplayName xml:lang="en">Maritime
              Information Sharing Environment</md:OrganizationDisplayName>
          <md:OrganizationURL xml:lang="en">http://www.mda.gov
          </md:OrganizationURL>
       </md:Organization>
       <md:ContactPerson contactType="technical">
          <md:Company>SPAWAR Systems Center Pacific</md:Company>
```

```
         <md:GivenName>Olithia</md:GivenName>
         <md:SurName>Strom</md:SurName>
         <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
         <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
      </md:ContactPerson>
   </md:EntityDescriptor>
   <md:EntityDescriptor entityID="https://mise.agencyone.gov/">
      <md:Extensions>
         <gfipm:EntityAttribute FriendlyName="COIIndicator"
            Name="mise:1.4:entity:COIIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue xsi:type="xs:string">True
            </gfipm:EntityAttributeValue>
         </gfipm:EntityAttribute>
         <gfipm:EntityAttribute FriendlyName="LawEnforcementIndicator"
            Name="mise:1.4:entity:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue xsi:type="xs:string">True
            </gfipm:EntityAttributeValue>
         </gfipm:EntityAttribute>
         <gfipm:EntityAttribute FriendlyName="PrivacyProtectedIndicator"
            Name="mise:1.4:entity:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue xsi:type="xs:string">True
            </gfipm:EntityAttributeValue>
         </gfipm:EntityAttribute>
         <gfipm:EntityAttribute FriendlyName="OwnerAgencyCountryCode"
            Name="mise:1.4:entity:OwnerAgencyCountryCode"
NameFormat="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
            <gfipm:EntityAttributeValue xsi:type="xs:string">USA
            </gfipm:EntityAttributeValue>
         </gfipm:EntityAttribute>
      </md:Extensions>
      <md:RoleDescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
         xsi:type="mise:MISEConsumerDescriptorType">
         <md:KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
               <ds:X509Data>
                  <ds:X509Certificate>
                     <!-- Base 64 encoded certificate embedded here
                        This is the client certificate which the trusted
                        system will present during SSL connection handshake.
                        The private key matching this certificate will also
                        be used by this trusted system for signing SAML
                        assertions.
                     -->
                  </ds:X509Certificate>
               </ds:X509Data>
            </ds:KeyInfo>
         </md:KeyDescriptor>
      </md:RoleDescriptor>
      <md:Organization>
         <md:OrganizationName xml:lang="en">Agency One
         </md:OrganizationName>
         <md:OrganizationDisplayName xml:lang="en">Agency
            One</md:OrganizationDisplayName>
         <md:OrganizationURL xml:lang="en">http://www.agencyone.gov/
         </md:OrganizationURL>
      </md:Organization>
      <md:ContactPerson contactType="technical">
         <md:Company>SPAWAR Systems Center Pacific</md:Company>
         <md:GivenName>Olithia</md:GivenName>
```

```
            <md:SurName>Strom</md:SurName>
            <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
            <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
        </md:ContactPerson>
    </md:EntityDescriptor>
    <md:EntityDescriptor entityID="https://mise.agencythree.gov/">
        <md:RoleDescriptor xsi:type="mise:MISEProviderDescriptorType"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
            <md:KeyDescriptor use="signing">
                <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                    <ds:X509Data>
                        <ds:X509Certificate>
                            <!-- Base 64 encoded certificate embedded here
                                This is the client certificate which the trusted
                                system will present during SSL connection handshake.
                            -->
                        </ds:X509Certificate>
                    </ds:X509Data>
                </ds:KeyInfo>
            </md:KeyDescriptor>
        </md:RoleDescriptor>
        <md:ContactPerson contactType="technical">
            <md:Company>SPAWAR Systems Center Pacific</md:Company>
            <md:GivenName>Olithia</md:GivenName>
            <md:SurName>Strom</md:SurName>
            <md:EmailAddress>olithia.strom@navy.mil</md:EmailAddress>
            <md:TelephoneNumber>619-553-0728</md:TelephoneNumber>
        </md:ContactPerson>
    </md:EntityDescriptor>
</md:EntitiesDescriptor>
```

4. MISE SAML Assertion Format

4.1. MISE SAML ASSERTION SPECIFICATION

SAML Assertions are used to convey user attribute information from information consumer systems to the ISI. This section contains normative language that describes MISE-specific requirements that apply to any SAML assertion generated by an information consumer system for use in MISE services. These requirements augment the SAML assertion format requirements that appear in the SAML 2.0 specification ([SAML2 Core]).

1. The **`<Assertion>`** element MUST be signed, MUST NOT be encrypted, and MUST be the root element. The **`<EncryptedAssertion>`** element is not used in MDA. Assertions are always signed and transmitted over SSL, but XML encryption is not used.
2. The **Version** attribute within **`<Assertion>`** MUST have "**2.0**" as its value.
3. The **`<Issuer>`** element within **`<Assertion>`** MUST be present, and its value MUST match the **entityID** in the trust fabric of the information consumer system initiating the service request sequence on behalf of a user.
4. The **`<ds:Signature>`** element MUST be present, and the **`<X509Certificate>`** within the **`<KeyInfo>`** element MUST be one of the signing certificates associated with the issuer in the trust fabric.
5. The **`<Subject>`** element MUST NOT be present.

6. The **<Conditions>** element MUST be present, and MUST contain the **NotBefore** and **NotOnOrAfter** attributes.
7. The **<AudienceRestriction>** element within **<Conditions>** MUST be present, and MUST contain an **<Audience>** element with the value **urn:mise:all**.
8. An **<Assertion>** element MUST NOT contain an **<AuthnStatement>** element.
9. An **<Assertion>** element MUST NOT contain an **<AuthzDecisionStatement>** element.
10. An **<Assertion>** element MUST contain exactly one **<AttributeStatement>** element.
11. The **<AttributeStatement>** element in an **<Assertion>** MAY contain one or more ‹Attribute› elements and MUST NOT contain any **<EncryptedAttribute>** elements.
12. Each **<Attribute>** element MAY contain application-level user attribute data corresponding to a MISE user attribute defined in [MISE Attributes].
13. If the **<Attribute>** element corresponds to a MISE user attribute defined in [MISE attributes], then the **Name** attribute within the **<Attribute>** element MUST contain the fully qualified formal name of the attribute as defined in [MISE Attributes].
14. Each **<Attribute>** element MUST contain one or more **<AttributeValue>** elements.
15. Each **<AttributeValue>** element MUST contain the following attribute name/value pairs:
    a. **xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**
    b. **xsi:type="xs:string"**
16. Each **<AttributeValue>** element MUST contain data corresponding to the value of the MISE user attribute represented by its enclosing **<Attribute>** element.

4.2. EXTENSION SCHEMA FOR ‹MD:ROLEDESCRIPTOR›

The diagram below contains the SAML Metadata extension schema for the ‹md:RoleDescriptor› element, which allows MISE roles and associated information to be stated explicitly in the trust fabric document.

```
<?xml version="1.0" encoding="US-ASCII"?>
<schema xmlns=http://www.w3.org/2001/XMLSchema
      xmlns:mise=http://mda.gov/standards/trustfabric/1.0
      xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      xmlns:ds=http://www.w3.org/2000/09/xmldsig#
      xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
      targetNamespace=http://mda.gov/standards/trustfabric/1.0
      elementFormDefault="unqualified"
      attributeFormDefault="unqualified"
      version="1.0">
   <import namespace="urn:oasis:names:tc:SAML:2.0:metadata"
schemaLocation="saml20/saml-schema-metadata-2.0.xsd"/>
   <import namespace="urn:oasis:names:tc:SAML:2.0:assertion"
schemaLocation="saml20/saml-schema-assertion-2.0.xsd"/>
   <import namespace=http://www.w3.org/2000/09/xmldsig#
      schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-
core-schema.xsd"/>

```

```
    <element name="MISEInfrastructureDescriptor"
type="mise:MISEInfrastructureDescriptorType"/>
    <element name="MISELoginService" type="md:EndpointType"/>
    <element name="MISELogoutService" type="md:EndpointType"/>
    <element name="MISESearchService" type="md:EndpointType"/>
    <complexType name="MISEInfrastructureDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType">
                <sequence>
                    <element ref="mise:MISELoginService"/>
                    <element ref="mise:MISELogoutService"/>
                    <element ref="mise:MISESearchService"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

    <element name="MISEConsumerDescriptor" type="mise:MISEConsumerDescriptorType"/>
    <complexType name="MISEConsumerDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType"/>
        </complexContent>
    </complexType>

    <element name="MISEProviderDescriptor" type="mise:MISEProviderDescriptorType"/>
    <complexType name="MISEProviderDescriptorType">
        <complexContent>
            <extension base="md:RoleDescriptorType"/>
        </complexContent>
    </complexType>
</schema>
```

4.3. SAMPLE SAML ASSERTION

The diagram below contains a sample SAML assertion, which provides a cryptographic guarantee of the identity of the trusted system asserting the user attributes and promising to handle the information in accordance with MISE guidelines.

```
<saml2:Assertion xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
        xmlns:xs=http://www.w3.org/2001/XMLSchema
        ID="_1025e5dabb24f891e338c4d38171982e" IssueInstant="2012-12-05T14:50:21.085Z"
        Version="2.0">
    <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://mise.agencyone.gov/</saml2:Issuer>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
            <ds:Reference URI="#_1025e5dabb24f891e338c4d38171982e">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature"/>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="xs"/>
                    </ds:Transform>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>qepBuvjTTzrg+I7YTHes8nxPFEY=</ds:DigestValue>
            </ds:Reference>
        </ds:SignedInfo>
```

```
      <ds:SignatureValue><!-- Base 64 encoded signature --></ds:SignatureValue>
      <ds:KeyInfo>
         <ds:X509Data>
            <ds:X509Certificate>
               <!-- Base 64 encoded certificate embedded here
                  This is the certificate of the information consumer system
                  which signed the assertion.
               -->
            </ds:X509Certificate>
         </ds:X509Data>
      </ds:KeyInfo>
   </ds:Signature>
   <saml2:Conditions NotBefore="2012-12-05T14:50:16.085Z" NotOnOrAfter="2012-12-
05T15:00:21.085Z">
      <saml2:AudienceRestriction>
         <saml2:Audience>urn:mise:all</saml2:Audience>
      </saml2:AudienceRestriction>
   </saml2:Conditions>
   <saml2:AttributeStatement>


      <saml2:Attribute FriendlyName="ElectronicIdentityId"
Name="gfipm:2.0:user:ElectronicIdentityId"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
         <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">eric.jakstadt@trustedfederal.com</saml2:AttributeValue>
      </saml2:Attribute>
      <saml2:Attribute FriendlyName="FullName" Name="gfipm:2.0:user:FullName"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
         <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">Eric G. Jakstadt</saml2:AttributeValue>
      </saml2:Attribute>
      <saml2:Attribute FriendlyName="CitizenshipCode"
Name="mise:1.4:user:CitizenshipCode" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
         <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">USA</saml2:AttributeValue>
      </saml2:Attribute>
      <saml2:Attribute FriendlyName="LawEnforcementIndicator"
Name="mise:1.4:user:LawEnforcementIndicator"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
         <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">true</saml2:AttributeValue>
      </saml2:Attribute>
      <saml2:Attribute FriendlyName="PrivacyProtectedIndicator"
Name="mise:1.4:user:PrivacyProtectedIndicator"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
         <saml2:AttributeValue xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="xs:string">true</saml2:AttributeValue>
      </saml2:Attribute>
   </saml2:AttributeStatement>
</saml2:Assertion
```

APPENDIX D - PUBLICATION INTERFACE SPECIFICATION

Maritime Information Sharing Environment (MISE)

Publication Interface Specification

Version 1.0

25 March 2013

1. Publication Overview

As discussed in the *National MDA Architecture*, information providers can choose between two integration approaches. In Figure 25, below, Information Provider System B chose to publish information to the Information Sharing Infrastructure (ISI) cache and delegate to the ISI the work of responding to access requests on behalf of users from information-consumer systems. This specification presents the details of the *Publication* interface used by a provider such as Figure 25's Provider B to keep the ISI cache up to date as a data set changes over time.

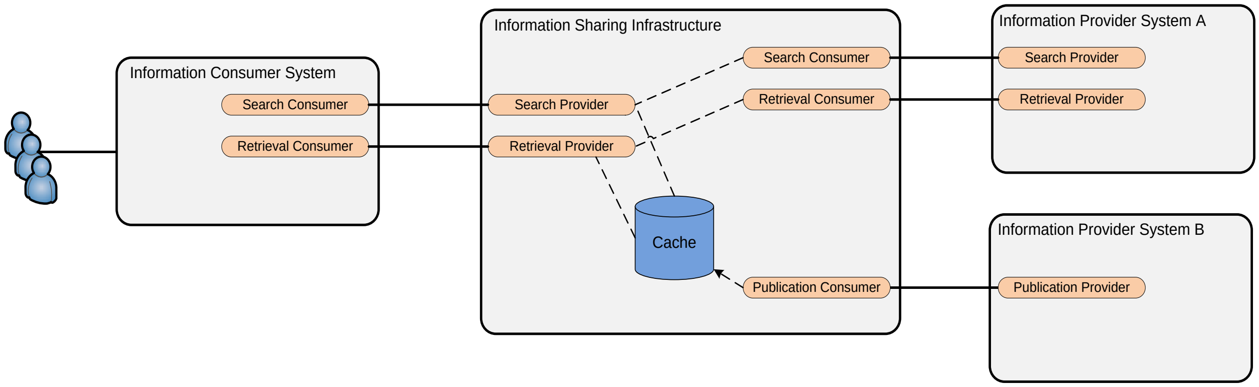


Figure 25 - Service Providers and Consumers

The publication interface follows the Representational State Transfer (REST) style. The ISI defines a URI endpoint for publication, and information-provider systems send HTTP requests and receive responses to URI paths beneath this URI endpoint. Details of these HTTP request and response messages are covered in Section 3.

All messages are authenticated and secured in the manner described in the *MISE Interface Security Specification*. For publication, the *Trusted System Authentication* portion of the interface security specification applies, but the *User Attribute Conveyance* portion does not apply since publication is not done on behalf of any individual user.

Each published data set must have an associated *Information Access Policy* (IAP). This IAP information is carried in each record published to the ISI. For more information on the IAP, see the *MISE Attribute Specification* and the *National MDA Information Exchange Package Documents (IEPD)*. The IAP is carried via attributes in the XML documents published to the ISI.

2. Message Flow Patterns

This section describes the sequence of HTTP request/response messages that is expected to occur during normal operation. These processes help keep the ISI cache up to date.

2.1. INITIAL FULL PUBLICATION

Before publishing a data set, an information provider will work with MISE management to configure the trusted system and data set at the ISI. This includes creating an IAP for the data set. Once these steps are complete, the information provider should do an initial full publication of the data set to the ISI, meaning all records that the information provider wishes to share should be sent to the ISI cache using the publication interface described herein.

The first step is to HTTP GET the version resource (see Section 3.1.1). If the version resource cannot be read, or if the version number is not one that the information provider is implemented to support, no further interaction with the interface should be taken.

After that, the information provider should HTTP PUT each individual shareable record (see Section 3.2.1). During this process, if the information provider fails to connect to the ISI, or if the ISI returns a status code indicating a server-side error (5xx), the information provider should periodically retry the PUT until it succeeds and contact the MISE helpdesk if errors persist.

If the information provider's back-end data store changes during the initial full publication, the provider system must track that and ensure that all shareable records have been successfully PUT to the ISI before the initial full publication process is considered complete.

Note that since the publication interface follows HTTP RESTful semantics, no harm is done if the information provider PUTs the same record more than once. The HTTP terminology for this characteristic is *idempotence*. Therefore, during any kind of error-recovery scenario, the provider is free to PUT the record if there is any question whether a previous PUT succeeded.

2.2. ONGOING UPDATING

After the initial full publication is complete, the information provider should update the ISI cache whenever any changes occur to the set of shareable records.

As with initial full publication, the information provider should periodically HTTP GET the version resource to confirm that the version of the publication interface is compatible. This does not necessarily need to be done before each record update, but it should be checked frequently—at least daily. The version resource is defined to support If-Modified-Since to make the check highly efficient.

Whenever any shared record is added or changed in the provider's back-end data store, the full record should be sent to the ISI using HTTP PUT (see Section 3.2.1). Whenever any shared record is deleted from the provider's back-end data store, the provider should use HTTP DELETE (see Section 3.2.2) to delete the record from the ISI cache. If the information provider fails to connect to the ISI, or if the ISI returns a status code indicating a server-side error (5xx), the information provider should periodically retry the PUT or DELETE until it succeeds and contact the MISE helpdesk if errors persist. Note that the HTTP DELETE method is also idempotent, so the provider is free to DELETE the record if there is any question whether a previous DELETE succeeded.

3. Resource Reference

This section presents details of resources defined as part of the publication interface and aspects of HTTP protocol usage that are important to interoperability.

All URIs are defined relative to ‹*BaseURI*›, which represents the HTTP endpoint of the publication interface at the ISI. The physical URI will be provided when the trusted system is integrated with the ISI. For example only, this specification uses https://isi.gov/publish as the base URI.

3.1. METADATA

3.1.1.INTERFACE VERSION

The version of the MISE publication interface described in this document is 1.0. Before interacting further, an information provider should GET the version resource to confirm that the ISI interface version matches what is expected by the information provider implementation.

Table 1 presents details of the HTTP GET request an information provider uses to retrieve the version resource and the possible responses, which may be received back from the ISI.

URI	‹*BaseURI*›/version	Example: https://isi.gov/publish/version
Method	GET	
Request Headers	Authorization	As described in the *MISE Interface Security Specification.*
	If-Modified-Since	Information provider system may send this header if it has previously read and cached this resource.
Request Content Type		Empty
Status Codes	200 (OK)	Successful. Response content as described below.
	304 (Not Modified)	The version resource has not been modified since the time specified in the If-Modified-Since request header.
	401 (Unauthorized)	No Authorization header or userid/password does not match any trusted system.
Response Headers	Last-Modified	
Response Content Type	application/xml; charset=UTF-8	Response content only returned if status code is 200.

Response Content	`<?xml version="1.0" encoding="UTF-8"?>`
	`<MISEInterface>`
	`    <Name>Publication</Name>`
	`<MajorVersion>1</MajorVersion>`
	`<MinorVersion>0</MinorVersion>`
	`</MISEInterface>`	

Table 1 – Version Resources

3.2. RECORD

The record resource represents an individual record as stored in the ISI cache.

The URI for an individual record is ‹*BaseURI*›/‹*IEPDName*›/‹*RecordID*›, where:

- ***‹IEPDName›*** matches one of the three currently defined national information products (noa, ian, pos).
- ***‹RecordID›*** is assigned by the information provider. It is unique to an individual record within the scope of the provider's data set.

For example, a full record URI may be similar to https://isi.gov/publish/noa/12345.

Before a record URI may be accessed, the ISI must be configured to accept a particular IEPD type from the provider. As discussed in the *MISE Interface Security Specification*, all HTTP requests to the ISI are authenticated, so the ISI knows which provider data set the record URI refers to—even though that information is not encoded within the URI. Records associated with a particular information provider cannot be accessed in any way through the publication interface by any other information provider.

3.2.1. ADD OR UPDATE A RECORD

Table 2 presents details of the HTTP PUT request an information provider uses to add or update a record and the possible responses, which may be received back from the ISI.

URI	‹*BaseURI*›/‹*IEPDName*›/‹*RecordID*›	Example: https://isi.gov/publish/noa/12345
Method	PUT	
Request Headers	Authorization	As described in the *MISE Interface Security Specification.*
Request Content Type	application/xml; charset=UTF-8	
Request Content	NIEM-M representation of record.	Must validate against the schema for *IEPDName* but need not include a schemaLocation attribute. The XML declaration should specify UTF-8 encoding.
Status Codes	201 (Created)	Record did not previously exist at ISI and was successfully added.
	204 (No Content)	Record previously existed at ISI and was successfully updated.
	400 (Bad Request)	Request content did not validate against the schema for *IEPDName*.
	401 (Unauthorized)	No Authorization header, or userid/password does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication

		of this IEPD.
Response Headers	Location	Only sent with status code 201. Matches record URI of request.
Response Content Type		Empty

Table 2 – Adding or Updating Resources

3.2.2. DELETE A RECORD

Table 3 presents details of the HTTP DELETE request an information provider uses to delete a record and the possible responses, which may be received back from the ISI.

URI	*<BaseURI>/<IEPDName>/<RecordID>*	Example: https://isi.gov/publish/noa/12345
Method	DELETE	
Request Headers	Authorization	As described in the *MISE Interface Security Specification.*
Request Content Type		Empty
Status Codes	204 (No Content)	Record was successfully deleted if it existed.
	401 (Unauthorized)	No Authorization header, or userid/password does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication of this IEPD.
Response Content Type		Empty

Table 3 – Deleting Records

3.2.3. PUBLISHING WITH A DATA SCOPE

For specific events and situations, the ISI provides the means for a data provider to specify a different level of data access. For instance, a data provider might allow temporary access to vessel position data for a wider range of data consumers during a hurricane. To publish data in a specific scope, the data provider must provide the **Scope** and **DataAttribute** parameters. A new record can be published with scope, or an existing record can be updated with a new scope.

URI	*<BaseURI>/<IEPDName>/<RecordID>?Scope=XXX&DataAttribute=YYYl&Releasable=B&Nation=NNN,MMM*	Example: https://isi.gov/publish/noa/12345?Scope=HurricaneKatrina&DataAttribute=COI&Releasable=F&Nation=USA
Scope	String defining the scope in which	HurricaneKatrina

	this record has modified data access. These are defined by the MISE Board for specific events	
DataAttribute	This is the modified data access attribute for that scope, as defined in the *MISE Attribute Specification*.	COI
Releasable	*Optional*. Boolean. Indicates whether the data is releasable within the scope.	T or F
Nation	*Optional*. Comma-separated list of ISO 3-letter country codes. Indicates which nations to which the data can be provide in this scope.	USA,CAN
Method	PUT	
Request Headers	Authorization	As described in the *MISE Interface Security Specification*.
Request Content Type	application/xml; charset=UTF-8	
Request Content	NIEM-M representation of record.	Must validate against the schema for *IEPDName* but need not include a schemaLocation attribute. The XML declaration should specify UTF-8 encoding.
Status Codes	201 (Created)	Record did not previously exist at ISI and was successfully added.
	204 (No Content)	Record previously existed at ISI and was successfully updated.
	400 (Bad Request)	Request content did not validate against the schema for *IEPDName*.
	401 (Unauthorized)	No Authorization header, or userid/password does not match any trusted system.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for publication of this IEPD.
Response Headers	Location	Only sent with status code 201. Matches record URI of request.
Response Content Type		Empty

Table 4 – Adding or Updating a Record with Data Scope

3.2.4. RECORD EXPIRATION

Records in the ISI cache will always be considered expired and be deleted after 30 days in the cache. Data providers can exercise more precise control over their records via the following means:

1. The DELETE interface described in 3.2.2, above.

18. Each IEPD includes the DocumentExpirationDate element. When set to a value less than 30 days, the record will be expired and deleted from the cache when that date is reached. If set to more than 30 days, it will be ignored and the record is deleted at 30 days.

APPENDIX E - SEARCH/RETRIEVE INTERFACE SPECIFICATION

Maritime Information Sharing Environment (MISE)

Search/Retrieve Interface Specification
Version 1.0

25 March 2013

1. Introduction

The Maritime Information Sharing Environment (MISE) is being established to allow secure sharing of unclassified information among partners within the MDA community of interest. The Information Sharing Infrastructure (ISI) is the information and request broker for the MISE.

This document describes the data-consumer facing representational state transfer (REST) architecture providing search and retrieve (SR) functionality for the MISE. By conforming to this interface, the ISI provides data consumers with the ability to find and retrieve the right information at the right time, based on the needs, rights, and authorities of the user and the organizations requesting the information. This document outlines the base level of functionality for the SR interface. The REST interface is divided logically into two parts. The search interface provides query endpoints and an associated set of query arguments, returning a list of results and summaries of each result. The retrieval interface provides the ability to access full records. The two interfaces work together to provide access to all information in the ISI.

All interactions with the SR interface are secured as described by the *MISE Interface Security Specification*. For search and retrieve operations, both the *Trusted System Authentication* and the *User Attribute Conveyance* portions of the interface security specification apply. All records available on the ISI are formatted using NIEM-Maritime, as described in the *National MDA Architecture Information Exchange Package Documents (IEPD)*. For further details on interfacing with the SR interface, see the *MISE Implementation Guide.*

2. General Consumer Search Interface

2.1. URL Structure and Query Results

For the purposes of this document, the ISI is assumed to be accessible at the global uniform resource identifier (URI) https://mise.mda.gov/services/MDAService, provided as an example base URL. SR provides a series of URL endpoints that provide global query functionality and focus-area-specific (or IEPD-specific) queries. The query URLs have the following form:

- https://mise.mda.gov/services/MDAService/search/<iepdname>?=

The <iepdname> takes the form of one of the message types provided by the ISI. Currently, these are:

1. noa (Notice of Arrival)
19. ian (Indicator and Notification)
20. pos (Position)
21. loa (Levels of Awareness)

Each URI in association with an IEPD name queries a single record type. For example, https://mise.mda.gov/services/MDAService/search/noa?= provides the notice-of-arrival specific query endpoint. Using this scheme, further endpoints can be added for specific queries as new record types are defined for new focus areas.

All queries to the search interface return an Atom feed that contain summaries of the record or records matching that query. The following listing shows an example feed returned from a search for Position interface. The search that returns this result takes the following form:

```
https://mise.mda.gov/services/MDAService/search/track?start=2012-08-
19T11:40:00&end=2013-12-30T11:40:00&ulat=3.75&ulng=-2.0&llat=-2.75&amp;llon=3.0
```

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:dc="http://purl.org/dc/elements/1.1/">
    <title>Query Response for Entity: https://mise.agencyone.gov/</title>
    <link rel="alternate"
href="https://mise.mda.gov/services/MDAService/search/track?start=2012-08-
19T11:40:00&amp;end=2013-12-30T11:40:00&amp;ulat=3.75&amp;ulng=-2.0&amp;llat=-
2.75&amp;llon=3.0" />
    <subtitle>Query Response from the Maritime Information Sharing
Environment</subtitle>
    <id>TRACK-9991956437ffa1c06327e0d98977989f4f7c6f46b1f3c52e3ac11d04d45273c3</id>
    <entry>
        <title>TRACK</title>
        <![CDATA[<link rel=”alternate”
href=https://mise.mda.gov/services/MDAService/retrieve/ian?entityid=https%3A%2F%2Fmise
.agencyone.gov%2F&recordid=79869882775656568/>]]>
        <author>
            <name>DCMP</name>
        </author>
        <id>8397109112108101736578</id>
        <updated>2012-12-31T19:28:08Z</updated>
        <summary type="text/xml">
            <posex:Message
xmlns:posex="http://niem.gov/niem/domains/maritime/2.1/position/exchange/3.2"
                mda:securityIndicatorText="LEI" mda:releasableNationsCode="USA"
mda:releasableIndicator="true"
                xmlns:m="http://niem.gov/niem/domains/maritime/2.1"
                xmlns:mda="http://niem.gov/niem/domains/maritime/2.1/mda/3.2"
                xmlns:nc="http://niem.gov/niem/niem-core/2.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
                xmlns:ism="urn:us:gov:ic:ism"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <mda:Vessel>
                    <m:VesselAugmentation>
                        <m:VesselIMONumberText>IMO0000001</m:VesselIMONumberText>
                        <m:VesselName>MV Example</m:VesselName>

<m:VesselNationalFlagISO3166Alpha3Code>USA</m:VesselNationalFlagISO3166Alpha3Code>
                    </m:VesselAugmentation>
                </mda:Vessel>
                <mda:Position>
                    <m:LocationPoint>
                        <gml:Point gml:id="tp1">
                            <gml:pos>-1.0 -1.0</gml:pos>
                        </gml:Point>
                    </m:LocationPoint>
                    <mda:PositionSpeedMeasure>
                        <nc:MeasureText>12</nc:MeasureText>
                        <nc:SpeedUnitCode>kt</nc:SpeedUnitCode>
                    </mda:PositionSpeedMeasure>
                    <mda:PositionCourseMeasure>
                        <nc:MeasureText>180</nc:MeasureText>
                        <m:AngleUnitText>deg</m:AngleUnitText>
                    </mda:PositionCourseMeasure>
                    <mda:PositionHeadingMeasure>
                        <nc:MeasureText>180</nc:MeasureText>
                        <m:AngleUnitText>deg</m:AngleUnitText>
                    </mda:PositionHeadingMeasure>
                    <mda:PositionNavigationStatus>
                        <nc:StatusText>Under way using engines</nc:StatusText>
                    </mda:PositionNavigationStatus>
                    <mda:PositionDateTime>
                        <nc:DateTime>2011-11-30T00:00:00Z</nc:DateTime>
                    </mda:PositionDateTime>
                </mda:Position>
            </posex:Message>
        </summary>
        <dc:creator>DCMP</dc:creator>
    </entry>
</feed>
```

In the Atom feed resulting from the query, four elements should be noted. First, the ‹link› element at the top level of the feed echoes back the query URL that returned this feed. Each individual record is content in an ‹entry› element. The ‹id› element supplies the unique ID of the record on the ISI. The ‹link› element provides the retrieval URL for the record on the ISI. As long as that record is available on the ISI, the link will provide access to it. The ‹summary› element contains the summary elements of the record. **Note that the ‹link› elements have replaced non-XML characters with their entity representations, so that the XML content is valid.**

All the records that match the query are returned, with one ‹entry› element for each record, up to the result size limit. See Sections 2.6 and 2.7 for more details on the headers and large result-set transfers.

The ISI search interface also provides utility endpoints:

- https://mise.mda.gov/services/MDAService/search/‹iepdname›/documentation: Provides the IEPD for that focus area, allowing new consumers to download the schema and other documentation for a record type representation.
- https://mise.mda.gov/services/MDAService/search/‹iepdname›/rendering: Provides rendering stylesheets to display the NIEM-M XML as HTML or other formats. Internally, the ISI will use this endpoint when a consumer provides an HTTP-Accept header other than the standard NIEM-M XML.
- https://mise.mda.gov/services/MDAService/specification: Provides a link to the current version of this document, describing the ISI REST search/retrieve interface.

2.2. PROTOCOL, SESSIONS, AND SECURITY

All interactions with the ISI are done over Secure Socket Layer/Transport Layer Security (SSL/TLS) connections. SSL/TLS are enforced to protect information in transit and session cookies. Each client querying against the SR interface is authenticated by the ISI and is provided with a limited-duration session cookie, which must be supplied in the header of following requests. The method for authentication is discussed the *MISE Interface Security Specification*.

2.3. SEARCH OPERATION

URI	*https://mise.mda.gov/services/MDAService/search/<iepdname>/?=*	Example: https://mise.mda.gov/services/MDAService/search/noa?VesselName=Enterprise
Method	GET	
Request Headers	Authorization	As described in the *MISE Interface Security Specification.*
Request Content Type	Empty	
Status Codes	200 (Success)	Returns the Atom feed with the results of the search.
	400 (Bad Request)	Request was not formatted correctly.
	401 (Unauthorized)	No Authorization header.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for search of this IEPD.
Response Headers	Empty	
Response Content Type	Atom feed with search results.	

Table 1 – Search Operation Detail

2.4. QUERY PARAMETERS

All IEPD search endpoints support a common set of query parameters, shown in Table 2.

Parameter Name	Value Type	Comment
VesselName	String	
VesselSCONUMText	String	
VesselMMSIText	String	
VesselIMONumberText	String	
VesselNationalFlag	String	
VesselClassText	String	
VesselCallSignText	String	
VesselHullNumberText	String	
VesselOwnerName	String	
VesselCategoryText	String	
InterestCategoryCode	String	
InterestLevel	String	
PortNameText	String	

InterestType	String	
InterestCategory	String	
InterestLevel	String	
InterestStart	ISO8601 DateTime	GMT
InterestEnd	ISO8601 DateTime	GMT
ArrivalPort	String	
ExpectedArrivalTime	ISO8601 DateTime	GMT
start	ISO8601 DateTime	GMT
end	ISO8601 DateTime	GMT
ulat	WGS84 upper left latitude of a bounding box	
ulng	WGS84 upper left longitude of a bounding box	
llat	WGS84 lower right latitude of a bounding box	
llng	WGS84 lower right longitude of a bounding box	
eid	String	The **eid** parameter allows a consumer to specify a specific data provider. This field must contain the entity ID of a provider system from the Trust Fabric document. This is only used in addition to other parameters, to narrow the returned results.

Table 2 – Common Query Arguments

These parameters can be combined to query for any of the record types.

- https://mise.mda.gov/services/MDAService/search/noa?VesselName=Atlantic%20Light&PortNameText=Miami

For **start**, **end**, and other dates, the representation is a GMT IS08601 datetime indicating a time period within which to return new or updated records. When either **start** or **end** date is specified, the time period is unbounded, but still a valid query. For instance, if only **start** is specified, there is no **end** time on the query:

- https://mise.mda.gov/services/MDAService/search/pos?VesselName=Atlantic%20Light&start=2012-02-29T00:00:00Z

Refer to the *MISE Implementation Guide* for examples, queries, and use cases.

2.5. SCOPE

For specific events and situations, the ISI provides an additional set of query parameters that can be used to query for messages within a specific scope. When applied by an information publisher, the scope parameters modify the entitlements for a record for the duration of the scope.

Parameter Name	Value Type	Comment
Scope	String	

For example, consider a Search query for all notice of arrival messages inbound to the Port of Miami during Hurricane Katrina:

- https://mise.mda.gov/services/MDAService/search/noa?PortNameText=Miami&scope=HurricaneKatrina

Valid scope values can be found on the National MDA Architecture website.

2.6. HEADER INFORMATION

The ISI SR interface uses the following HTTP headers for specific purposes. All other HTTP headers should be interpreted according to the HTTP 1.1 standard.

- Accept: By default, if no Accept is specified, the summary feed will be returned in the Atom format as specified above. However, the following Accept headers may be supplied, resulting in translated feeds:
- application/kml – returns the Atom feed translated into a KML overlay.
- Transfer-Encoding: For large requests, the server may respond using HTTP 1.1 chunked transfer. When this happens, the Transfer-Encoding HTTP response header is set in place of the Content-Length header, which the protocol would otherwise require. Because the Content-Length header is not used, the ISI does not need to know the length of the content before it starts transmitting a response to the client. The ISI can begin transmitting responses with dynamically generated content before knowing the total size of that content. The size of each chunk is sent right before the chunk itself so that a client can tell when it has finished receiving data for that chunk. The data transfer is terminated by a final chunk of length zero.

2.7. ERROR CODES

The ISI SR interface uses the following HTTP response codes for specific purposes. All other response codes should be interpreted according to the HTTP 1.1 standard.

- 413: If the resulting record set matching a query is too large for efficient transfer, even via HTTP chunked transfer, the ISI returns the 413 error code, indicating that the response set is too large. In this situation, the client should retry the query with a more restrictive set of parameters.

3. Focus-Area Specific REST Parameters

This section describes the more specific parameters to query information from each focus area. The URIs below follow the same format as the general query interface, with additional parameters described below to provide specific information.

3.1. NOTICE OF ARRIVAL

Parameter Name	Value Type	Comment
NoticeType	String	
NoticeTransactionType	String	
CDCCargoDeclared	Boolean	T or F

3.2. VESSEL POSITION

Parameter Name	Value Type	Comment
num	Integer	The **num** parameter is applied to queries for Position information. It specifies how many previous positions should be returned in reverse date order. The default number is 5.

3.3. INDICATORS AND NOTIFICATIONS

Parameter Name	Value Type	Comment
ActivityName	String	
ActivityStart	ISO8601 DateTime	GMT
ActivityEnd	ISO8601 DateTime	GMT

3.4. LEVELS OF AWARENESS

Parameter Name	Value Type	Comment
LevelOfAwarenessCode	Integer	This parameter is applied to queries specifically for the Levels of Awareness IEPD type. This specifies the LOA level, 1-4 that should be returned. If not specified, all matching LOA messages will be returned.

4. Retrieve Interface

In each Atom feed returned, the ‹link› element for each entry points to the canonical URI that represents that record. As long as the ISI can reference that record, the ID-specific URI will point to the same record. If the record is no longer accessible, the ISI returns an HTTP 404 status code, indicating that that record is no longer available. Each record URI takes the form https://mise.mda.gov/services/MDAService/retrieve/‹type›?entityid=‹eid›&recordid=‹id›. The ID is the same as the ‹id› element in the ‹entry›. The ‹eid› references the data provider system. This value is the Entity ID from the Trust Fabric document. The ‹type› variable is the focus area, such as *pos* for position reports. **Note that the ‹eid› parameter value URL-encoded. For example, the entity ID http://agencyone.gov would be represented in the URL as https%3A%2F%2Fmise.agencyone.gov.**

4.1. RETRIEVE OPERATION

URI	https://mise.mda.gov/services/MDAService/retrieve/‹type›?entityid=‹eid›&recordid=‹id›.	Example: https://mise.mda.gov/services/MDAService/retrieve/track?entityid=agencyone.gov&recordid=8397109112108101736578
Method	GET	
Request Headers	Authorization	As described in the *MISE Interface Security Specification.*
Request Content Type	Empty	
Status Codes	200 (Success)	Returns the NIEM-M formatted record.
	400 (Bad Request)	Request was not formatted correctly.
	401 (Unauthorized)	No Authorization header.
	403 (Forbidden)	Authenticated information provider is not configured at ISI for search of this IEPD.
Response Content Type	NIEM-M Record	

Table 3 – Retrieve Operations Detail

4.2. SCOPE

For specific events and situations, the ISI provides an additional set of query parameters that can be used to query for messages within a specific scope. When applied by an information publisher, the scope parameters modify the entitlements for a record for the duration of the scope.

Parameter Name	Value Type	Comment
Scope	String	

Scope must be applied to the retrieve query, similar to the search query:

- https://mise.mda.gov/services/MDAService/retrieve/track?entityid=agencyone.gov&recordid=8397109112108&scope=HurricaneKatrina

APPENDIX F - GOVERNANCE MANUAL

Maritime Information Sharing Environment (MISE)

Governance Manual

Version 1.0

1. Introduction

1.1. BACKGROUND

The National Maritime Domain Awareness (MDA) Concept of Operations describes an end-state in which any authorized user is able to access the information they require through seamless validation of the needs, rights and authorities of the user balanced against the permissions or restrictions established by organizations publishing the information. The Maritime Information Sharing Environment (MISE) provides that environment as a national capability to share and search maritime information across organizational boundaries relevant to the maritime domain.

Value to the Maritime Community:

- **User Convenience** – Users can access multiple services using a common set of standardized security credentials making it easier to sign on, access applications, and manage account information.
- **Interoperability** – MISE specifies common security standards and framework so applications can adopt interoperable security specifications for authentication and authorization.
- **Cost-Effectiveness** – MISE facilitates information sharing by using standardized Extensible Markup Language (XML)-based credentials that include information about each user's identity and privileges. This reduces the cost and complexity of identity administration required to access applications and vet users.
- **Privacy** – MISE will reduce the propagation of personally identifiable information (PII), reduce redundant capture and storage of PII, and depersonalize data exchanges across domains using privacy metadata.
- **Security** – the MISE model will improve the security of PII and data in participant organizations' applications by providing a standardized approach to online identities between agencies or applications.

Contents

The MISE Governance Manual defines the governance structure for MISE, including the parties that play a role in the governance structure (e.g. Board of Directors, MISE Management, information provider representatives, and information consumer representatives) and the decisions to be made by each party.

Target Audiences

The target audience for this document includes managers and technical representatives of prospective MISE participant organizations who are planning to implement a service within the MISE. It also includes vendors, contractors, and consultants who are required to establish technical interoperability with National Information Exchange Model - Maritime (NIEM-M) standards as part of a contracted project or product implementation.

2. MISE Governance Structure

The MISE governance structure will consist of the MISE Board of Directors (BOD), MISE Management, and representatives (Information Providers and Information Consumers) from various organizations participating in the MISE Configuration Control Board (CCB). The governance structure is captured in the figure below.

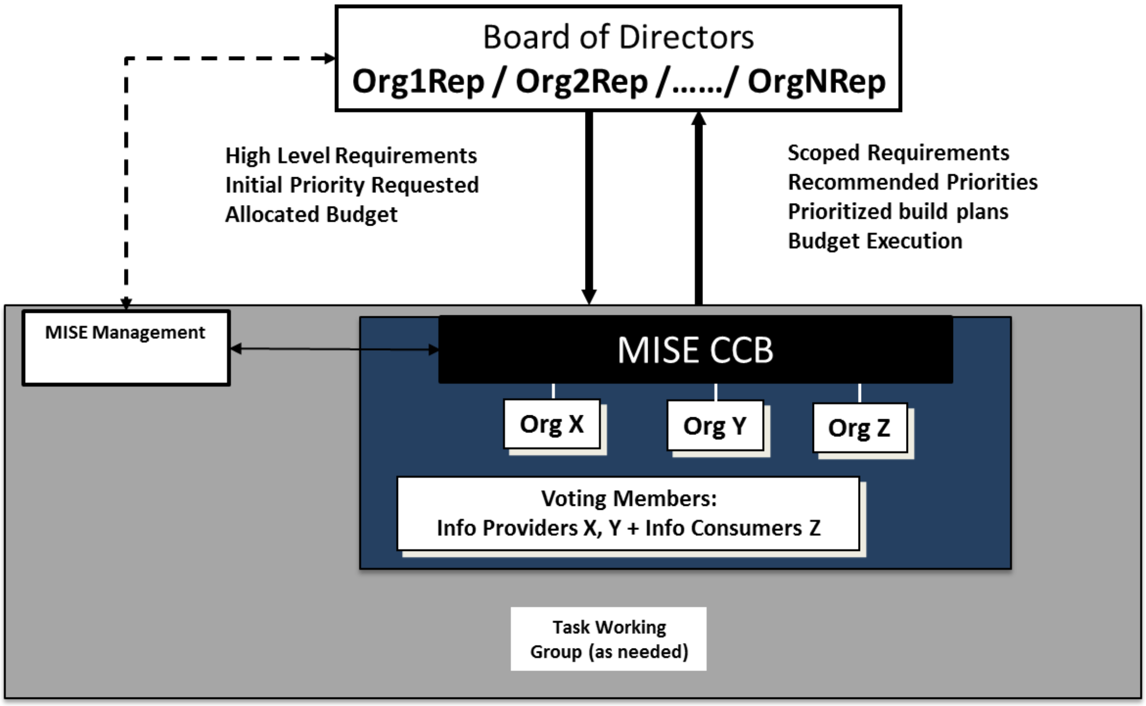


Figure 26 MISE Governance Structure

The specific organizations and individuals that comprise the BOD, MISE Management and CCB will evolve over time, so are not specified in this document.

3. Board of Directors

The MISE BOD is the executive-level body with representation from primary stakeholders that guides the MISE and is the final authoritative body to make decisions for the environment.

3.1. RESPONSIBILITIES

The BOD provides executive oversight of the MISE and decides on matters that are beyond the authority delegated to the MISE CCB. This includes, but is not limited to:

- Provide high level requirements
- Allocate the MISE budget
- Provide prioritization and approval for execution of CCB recommendations
- Approval authority for any modification to Trusted System Agreements

- Approval authority for any changes to any guidelines, standards, or documents of the MISE
- Approval authority for any changes to the governance structure
- Approval authority for membership policy, membership requests, membership suspension or revocation, and any changes to the MISE systems and security policies
- Identify potential new partners to join the environment

3.2. MEETINGS

The BOD shall meet quarterly to discuss general business and other matters that may arise. On an as needed basis, a BOD member may call an emergency meeting to discuss and decide matters that need immediate attention. To call an unscheduled meeting, the board member requesting the meeting must notify the other board members via written or electronic communication and obtain concurrence of no less than 50 percent of the Board to meet.

Regularly scheduled and emergency meetings can occur in person or via teleconference, and require a quorum of the Board of Directors to proceed. If a quorum is not present, the meeting shall be rescheduled to the next date when a quorum can be present. Emergency meetings will have no effect on the next regularly scheduled meeting date.

4. MISE Configuration Control Board

The CCB has responsibility for reviewing change requests for MISE exchange standards, services and processes, and forwarding recommendations to the BOD for approval decisions. The CCB is composed of representatives from information provider organizations, information consumer organizations, and MISE management.

4.1. RESPONSIBILITIES

The role of the CCB is to provide a common forum where information providers, information consumers and MISE management can jointly recommend MISE changes and enhancements, and forward those recommendations to the BOD. Specific responsibilities include:

- Review Change Requests (CR)
- Identify focus areas for new Information Exchange Package Documentation (IEPD)
- Provide recommendations to improve the execution of MISE
- Analyze recommendations for changes to any guidelines, standards, or documents of the MISE
- Analyze recommendations of any changes to the governance structure
- Analyze recommendations for membership policy, membership suspension and revocation, and any changes to the MISE systems and security policies
- Approve the formation of Task Working Groups (TWG) to target specific tasks that must be completed

- Approve the dissolution of TWG when tasks are completed
- Implement the MISE Work Flow process as shown in the diagram below.

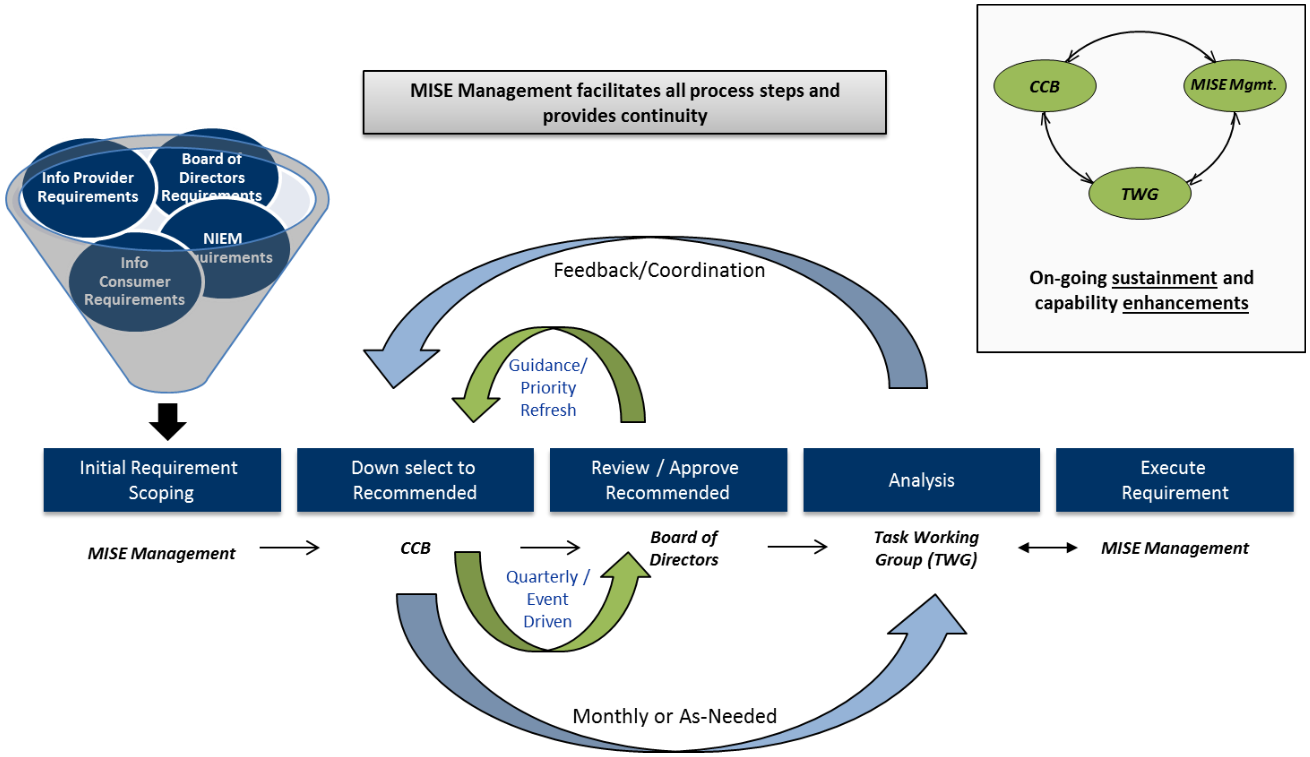

MISE Management facilitates all process steps and provides continuity
CCB
MISE Mgmt.
TWG
On-going sustainment and capability enhancements
Info Provider Requirements
Board of Directors Requirements
NIEM Requirements
Info Consumer Requirements
Feedback/Coordination
Guidance/ Priority Refresh
Initial Requirement Scoping
Down select to Recommended
Review / Approve Recommended
Analysis
Execute Requirement
MISE Management
CCB
Board of Directors
Task Working Group (TWG)
MISE Management
Quarterly / Event Driven
Monthly or As-Needed

Figure 27 MISE Work Flow

4.2. MEETINGS

The CCB shall meet monthly to discuss Change Requests, provide or analyze recommendations, and other matters that may arise. On an as needed basis, the CCB may call an emergency meeting to discuss and decide matters that need immediate attention. Any unscheduled meeting must be approved by MISE management.

Regularly scheduled and emergency meetings can occur in person or via teleconference. If a quorum is not present, the meeting shall be rescheduled to the next date when a quorum can be present. Emergency meetings shall have no effect on the next regularly scheduled meeting date.

5. MISE Management

MISE Management will facilitate the identification, collaboration, management, movement, and processing of maritime information by leveraging the MISE. The following are the objectives of MISE Management:

1. Lead the NIEM-M domain to standardize sharing of common maritime information with interagency and international partners

2. Provide standards-based environment to facilitate secure, seamless access to unclassified maritime information among trusted systems
3. Manage day-to-day operations of the MISE

5.1. RESPONSIBILITIES

The MISE Management will be responsible for the day-to-day operations of the MISE. This includes, but is not limited to, the following:

- Facilitate CCB Meetings
- Execute approved CCB requirements
- Maintain NIEM-M domain model and exchange standards
- Manage MISE Help Desk
- Assist information providers in developing and maintaining Information Access Policies (IAPs)
- Certify new Trusted Systems within the MISE
- Maintain and disseminate the Trust Fabric

6. Information Providers/Consumer Representatives

Information Providers / Information Consumers Representatives participate in the MISE with an approved trusted system adhering to the unclassified maritime information sharing environment rules established by MISE management.

7. Responsibilities of the MISE Governing Bodies

The responsibilities of the MISE are listed below in this section.

7.1. POLICY

A membership policy for the MISE will be established by the MISE Management and approved by the BOD. Any subsequent changes to the policy will require BOD approval.

7.2. APPROVAL

The BOD has the authority to approve new members; MISE Management has the authority to sign any service Level Agreements on behalf of the MISE. Any modifications to the standard agreement or assignments of these agreements will require the BOD approval.

7.3. Membership Suspension

If suspicion exists that a member has violated any of the MISE provisions, standards, policies, or procedures that threatens the integrity of the MISE, then MISE Management may suspend membership for up to 15 days. A longer suspension may be imposed by the BOD.

7.4. Audit / Investigate

The MISE has the right to audit the MISE-related activities of any member across the MISE. In the case of an alleged breach of MISE policy, the MISE Management will perform a basic audit and investigation. If there is initial validation of the concern based on this initial investigation, the member can be required to provide a third-party audit to show that their procedures adhere to MISE rules.

7.5. Membership Revocation

The decision to revoke any member's membership without cause will be in the sole discretion of the BOD and will require 60 days' notice to such member. If any member decides to cancel their membership, they may do so upon 60 days' notice to the MISE Management. The MISE Management will be responsible for periodically informing the BOD of the cancellation of any memberships.

8. Conflict Resolution

The MISE members agree that any dispute or conflict will be brought to the BOD, via MISE Management, for resolution. All decisions made by the BOD in resolution of a conflict will be deemed final upon notice to the parties involved. Unless otherwise stipulated in the resolution description, the decision will be implemented in 30 days from initial notice.

8.1. Disputes Among MISE Members

Disputes among MISE members shall be resolved via the following process. When a dispute occurs among members, one or more members may notify the MISE Management of the dispute in writing. Upon receiving written notification of the dispute, the MISE Management may, at its discretion, take the necessary steps to investigate the dispute. After investigating the dispute, the MISE Management may either render a decision to resolve the dispute or submit the issue to the BOD for a vote. The MISE Management must complete this process no later than 60 days after receiving written notification of the dispute.

8.2. Disputes between Members and the MISE Management

When a dispute occurs between any of the MISE members and the MISE Management, one or more parties of the dispute may notify the BOD of the dispute in writing. Upon receiving written notification of the dispute, the Director in receipt of the written notification must forward the notification to all members of the BOD. The BOD may, at its discretion, take the necessary steps to investigate and render a decision on the dispute. The MISE Board of Directors must complete this process no later than 60 days after receiving written notification of the dispute.

8.3. END USER CONFLICT

Any end-user conflict will be addressed solely by the Information Provider / Information Consumer Representative to which the user subscribes, and cannot be appealed to any other MISE member.

9. Core MISE Governance Documents

The operation of the MISE is governed by this MISE Governance Manual. This document provides the foundation for governing and operating the MISE.

Further details regarding the required processes to manage the MISE environment are described in the Appendices.

10. Glossary

Attributes	Characteristics of a persona that defines a user in a particular role (sent by a trusted system to the ISI)
Board of Directors	The Board of Directors decides on any matters that fall outside of the role of the MISE Management and provide general guidance. Some of the responsibilities include approval of any modification to standard agreements, documents, or governance structure.
MISE Management	The MISE Management is responsible for the day-to-day operations of the MISE.
Governance	Establishment of policies and continuous monitoring of the proper implementation by the members of the governing body of an organization
Information Consumer	A type of trusted system that authenticates users, using an internal or external identity provider, and passes information access requests and user attributes on behalf of the user to the ISI
Information Provider	A type of trusted system that provides both maritime information and corresponding Information Access Policy (IAP) to the ISI
Information Sharing Infrastructure (ISI)	Handles requests from the trusted systems on behalf of the users. Operating on the user attributes, it will make entitlement decisions, processing messages from the information providers based on their IAP and providing responses to the trusted system
Personally Identifiable Information (PII)	Information that can be used to uniquely identify, contact, or locate a single person or can be used with other sources to uniquely identify a single individual
Service Provider	An entity that provides services to other entities
Trusted System	Information provider and/or information consumer. It also passes information and IAP or information access requests and user attributes to the ISI. The trusted system relies on an internal or external identity provider to authenticate users.
User	Any person, organization, system, etc. that authenticates to the trusted system via an internal or external identity provider in order to access the ISI.

11. Acronyms

BOD	Board of Directors
CCB	Configuration Control Board
CR	Change Request
IAP	Information Access Policy
IEPD	Information Exchange Package Document
ISI	Information Sharing Infrastructure
MISE	Maritime Information Sharing Environment
MDA	Maritime Domain Awareness
NIEM-M	National Information Exchange Model – Maritime
PII	Personally Identifiable Information
TS	Trusted System
TWG	Task Working Group
XML	Extensible Markup Language

12. Request to Join Process

The Request to Join Process serves as a preliminary qualifications assessment for potential MISE Trusted Systems to join the environment. It provides an opportunity for potential MISE Stakeholders to learn about the environment and the BOD to determine applicability of the system for the environment. Figure C1 depicts the Request to Join Process. Table C1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Work with maritime community to identify new MISE Stakeholders	BOD	Discussions	Discussions displaying interest in MISE	At any time.
2	Review unsolicited requests by agencies to join MISE	MISE Management	Unsolicited request/interest	Discussions displaying interest in MISE	At any time.
2	Receive informational brief about MISE	MISE Stakeholder	MISE introduction brief	Presented MISE introduction brief	Within 20 days of initial interest
3	Notify Management of interested Stakeholder	BOD	Completed presentation of MISE introduction brief	Notified MISE Management of potential stakeholder	Within 5 days of giving MISE introduction brief
4	Schedule meeting with interested MISE stakeholder	MISE Management	Notification from MISE management of potential stakeholder	Scheduled meeting with potential stakeholder	Within 10 days of notification
5	Technical evaluation of readiness of system	MISE Stakeholder	Information provider survey	Completed information provider survey	Occurs during meeting previously scheduled
6	Compile recommendation report	MISE Management	Completed information provider survey	Completed recommendation report	Within 5 days of meeting with potential stakeholder
7	Review recommendation report	BOD	Completed recommendation	Analysis and Authorization Decision	Meeting of BOD (Quarterly or as needed)
8	Notified of rejection	MISE Stakeholder	BOD Decision	Notification to stakeholder of decision	Within 5 days of BOD Decision
9	Notified of approval	MISE Stakeholder	BOD decision	Notification to stakeholder of decision	Within 5 days of BOD Decision

Table C1. Request to Join Process

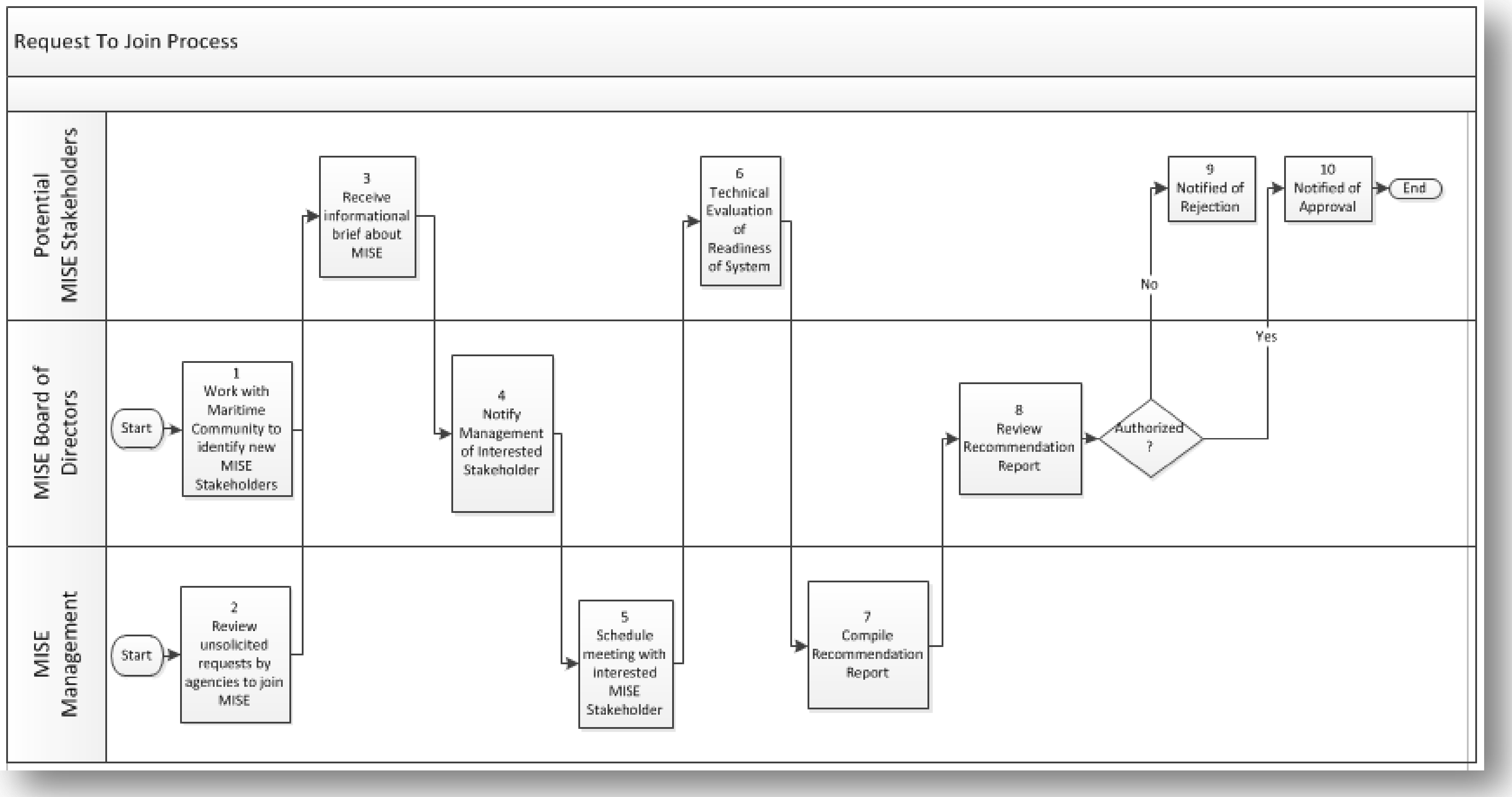

Request To Join Process
Potential MISE Stakeholders
MISE Board of Directors
MISE Management
Start
1 Work with Maritime Community to identify new MISE Stakeholders
Start
2 Review unsolicited requests by agencies to join MISE
3 Receive informational brief about MISE
4 Notify Management of Interested Stakeholder
5 Schedule meeting with interested MISE Stakeholder
6 Technical Evaluation of Readiness of System
7 Compile Recommendation Report
8 Review Recommendation Report
Authorized ?
No
Yes
9 Notified of Rejection
10 Notified of Approval
End

Figure C1: Request to Join Process

13. MISE Help Desk Process

The purpose of the Help Desk is to solve operational problems raised by end-users. As the MISE is comprised of many member organizations, each with its own local help desk resources, stakeholders must leverage these local resources to the maximum extent possible. The primary guiding principle in the design of the MISE help desk structure is that all problems SHOULD be solved as close to the user as possible, and with as little centralized effort as possible. Figure D1 depicts the Help Desk Process. Table D1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Experience IT Problem	MISE Trusted System User	Experienced IT problem	Experienced IT problem	At any time
2	Work with Local network Help Desk to Resolve Issue	User’s Network Help Desk	Experienced IT problem	Unresolved IT problem	At any time
3	Work with Trusted System Help Desk to resolve issue	MISE Trusted System	Unresolved It problem	Unresolved IT problem	At Any time
4	Submit issue through mda.gov	MISE Management	Unresolved IT problem	Resolved IT problem	Within 10 business days of notification
5	Work with Trusted System Help Desk to Resolve Issue.	MISE Management	Unresolved IT problem	Resolved IT problem	Within 10 business days of notification

Table D1. MISE Help Desk Process

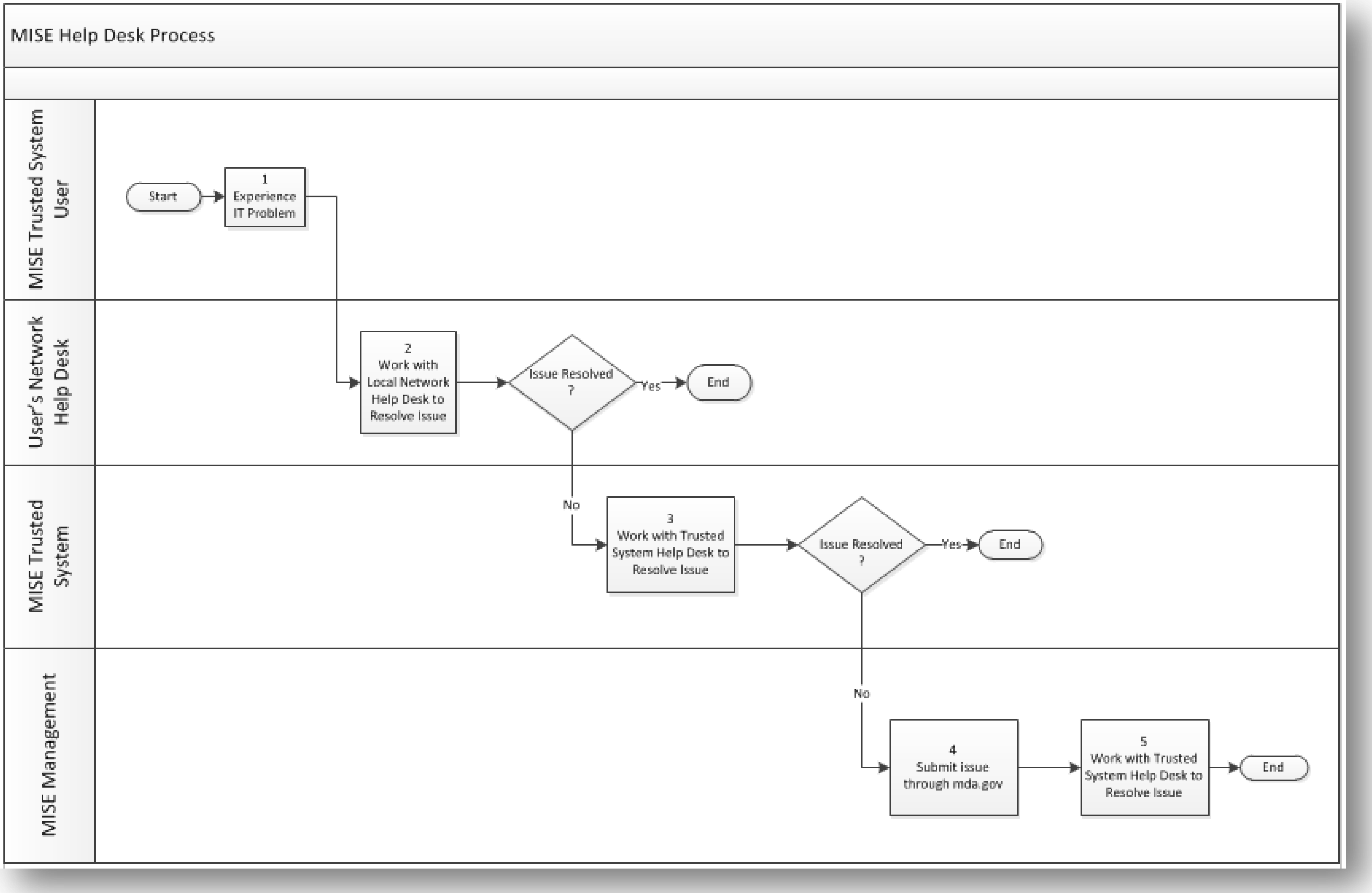

MISE Help Desk Process
MISE Trusted System User
User's Network Help Desk
MISE Trusted System
MISE Management
Start
1 Experience IT Problem
2 Work with Local Network Help Desk to Resolve Issue
Issue Resolved ?
Yes
End
No
3 Work with Trusted System Help Desk to Resolve Issue
Issue Resolved ?
Yes
End
No
4 Submit issue through mda.gov
5 Work with Trusted System Help Desk to Resolve Issue
End

Figure D1. MISE Help Desk Process

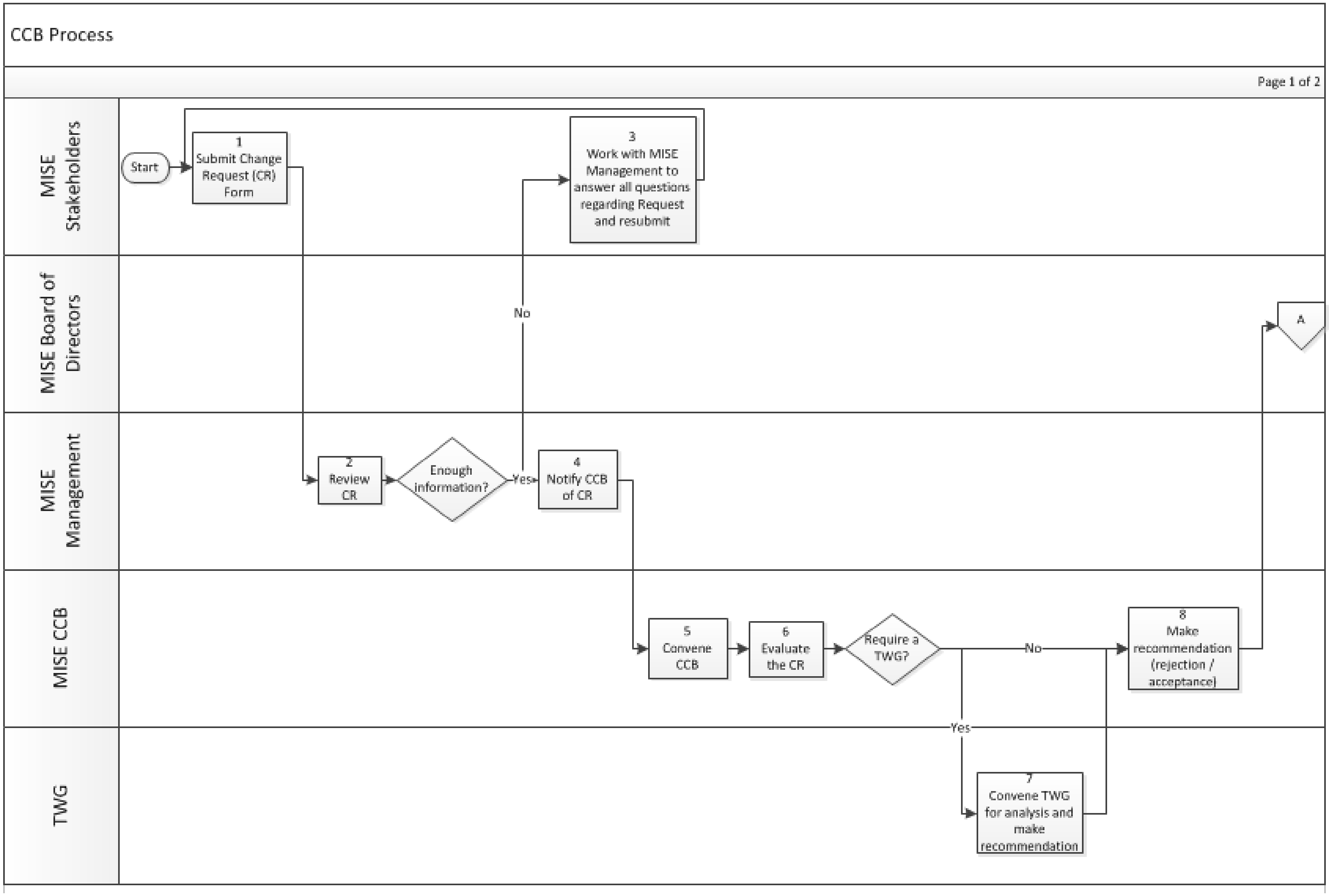
14. Configuration Control Board Process

The MISE CCB is responsible for controlling the MISE baseline, and evaluating and approving proposed changes. Figure E1 depicts the CCB Process. Table E1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Submit Change Request (CR) Form	MISE Stakeholder	CR Form Template	CR Form	At any time.
2	Review CR	MISE Management	CR Form	Completed CR Form	Within 10 working days of receipt of CR
3	Work with MISE Management to answer all questions regarding request and resubmit	MISE Management / MISE Stakeholder	Incomplete CR Form	Completed CR Form	Within 5 working days after review of CR
4	Notify CCB of CR	MISE Management	Completed CR Form	Confirmed notification to CCB	Within 15 days of receipt of CR
5	Convene the CCB	CCB	Completed CR Form	Scheduled CCB Meeting	At least quarterly or as needed
6	Evaluate the CR	CCB	Scheduled CCB Meeting Completed CR Form	CCB Meeting Minutes capturing discussion	Within 5 days of CCB Meeting
7	Convene TWG for analysis and make recommendation	TWG	Direction from CCB Completed CR Form	Recommendation / Rejection Report	Within 5 days of CCB Meeting
8	Make recommendation (rejection/approval)	CCB	CCB Meeting Minutes capturing discussion Completed CR Form TWG Recommendation/Rejection Report	CCB recommendation / rejection Cost estimate (if needed)	Quarterly
9	Review CR and recommendation	BOD	Scheduled BOD Meeting Completed CR Form CCB recommendation/rejection	BOD meeting minutes capturing discussion BOD cost estimate approval (if	Quarterly

			Cost estimate (if completed)	needed) BOD authorization / rejection	
10	Plan Update	CCB	BOD authorization BOD cost estimate approval (if needed)	Planned schedule Planned changes	Next CCB meeting
11	Close CR	CCB	BOD rejection	Closed CR Explanation of rejection	Within 5 days of decision
12	Notified of closure and explanation of rejection	MISE Stakeholder	Closed CR Explanation of rejection	MISE Stakeholder acknowledgement of rejection	Within 5 days of decision
13	Notified of approval and status update	MISE Stakeholder	Planned changes Planned schedule	MISE Stakeholder acknowledgment of approval	Within 5 days of decision
14	Make changes	MISE Management	Planned Schedule Planned Changes	Implemented changes and final Report	As needed dependent on tasking and resources available
15	Close CR	CCB	Implemented changes and final report	Closed CR	Within 5 Days of changes implemented and finalized
16	Notified of completion	Information Provider / Information Consumer	Closed CR	MISE Stakeholder acknowledgement of completion	Within 5 days of closure

Table E1. CCB Process


CCB Process
Page 1 of 2
MISE Stakeholders
MISE Board of Directors
MISE Management
MISE CCB
TWG
Start
1 Submit Change Request (CR) Form
3 Work with MISE Management to answer all questions regarding Request and resubmit
No
A
2 Review CR
Enough information?
Yes
4 Notify CCB of CR
5 Convene CCB
6 Evaluate the CR
Require a TWG?
No
Yes
8 Make recommendation (rejection / acceptance)
7 Convene TWG for analysis and make recommendation

1

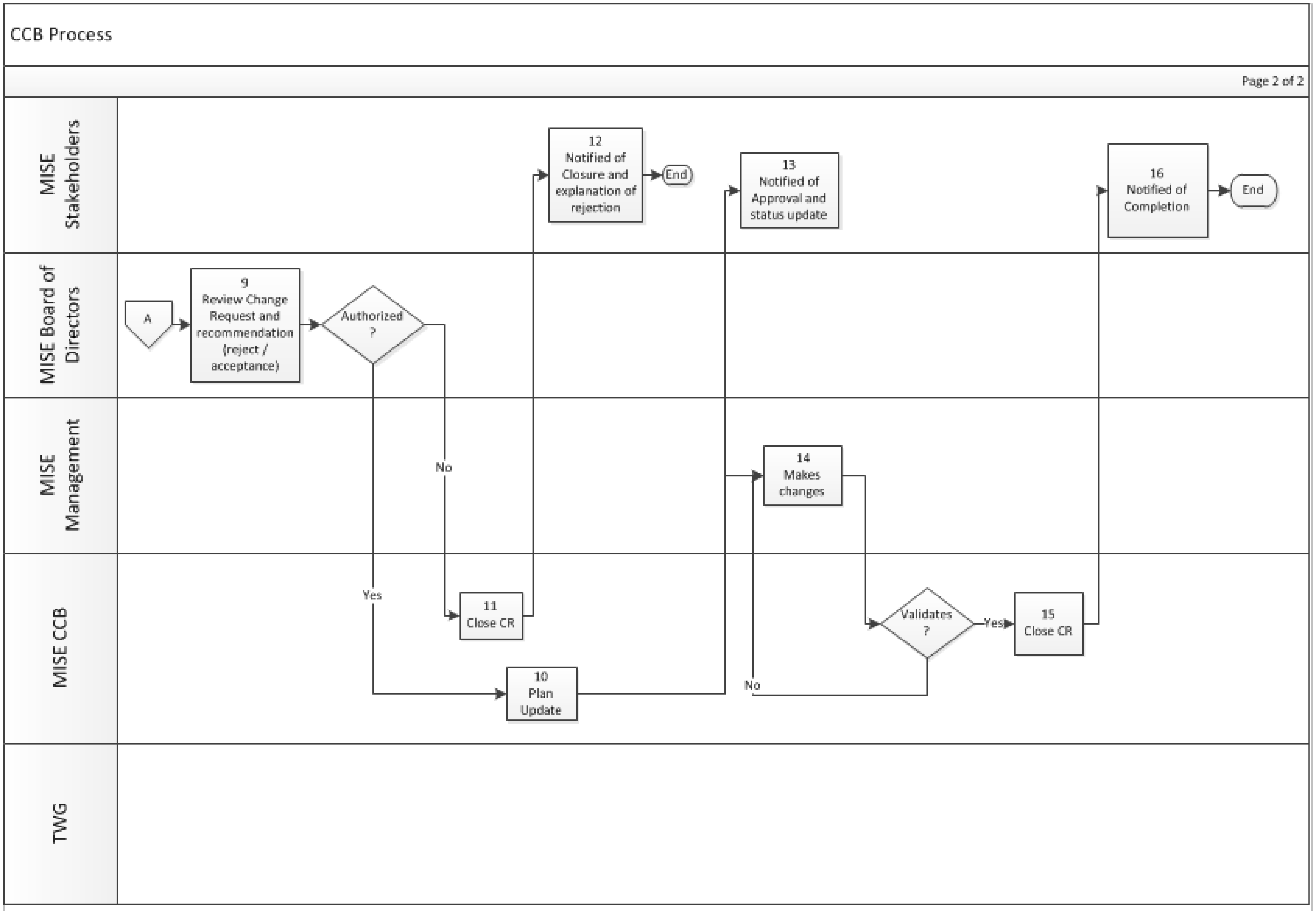

CCB Process
Page 2 of 2
MISE Stakeholders
MISE Board of Directors
MISE Management
MISE CCB
TWG
A
9 Review Change Request and recommendation (reject / acceptance)
Authorized ?
No
Yes
10 Plan Update
11 Close CR
12 Notified of Closure and explanation of rejection
End
13 Notified of Approval and status update
14 Makes changes
Validates ?
Yes
No
15 Close CR
16 Notified of Completion
End

Figure E1. CCB Process

15. Onboarding Process

Once a MISE Stakeholder is authorized to join the environment, the Information Provider/Information Consumer will complete the implementation in accordance with the MISE Implementation Guide. The Onboarding Process will allow the Trusted System to be connected to the MISE environment. Figure F1 depicts the Onboarding Process. Table F1 provides a detailed explanation of each step in the process.

#	Description	Actor	Input	Output	Time Frame
1	Complete Implementation	MISE Trusted System	Authorization to join MISE Environment	Completed Data Mapping Model Transforms (if needed) Attribute Mapping Developed IAP Implemented Required Services (Publication, Search, Retrieve, Security)	At any time.
3	Add Trusted System to the Test Trust Fabric	MISE Management	Procured Certificate / Key	Trusted System added to the Test Trust Fabric	Within 10 days of TS certificate in key store
4	Submit IAP	MISE Management	Trusted System added to the Trust Fabric	Implemented IAP	Within 10 days of TS certificate in key store
5	Issue HTTP PUT / HTTP GET	MISE Trusted System	Implemented IAP	Issued HTTP PUT/HTTP GET	Within 10 days of TS certificate of key store
6	Test Interoperability	MISE Management / MISE Trusted System	Issued HTTP PUT/HTTP GET	Successful Interoperability Test	Within 10 days of TS certificate of key store
7	Work with MISE Management to address interoperability issues	MISE Trusted System	Unsuccessful interoperability test	Identified and corrected issues	Within 5 days of unsuccessful interoperability test
8	Add Trusted System to	MISE Management	Successful interoperability test	Trusted System add to the Operational Trust Fabric	Within 5 days of successful interoperability test

	Operational Trust Fabric				
9	Troubleshoot	MISE Management	Unsuccessful interoperability test of Operational Trust Fabric	Identified and corrected issues Successful interoperability test of Operational Test Fabric	Within 5 days of unsuccessful test
10	Notified of successful Onboarding	CCB and BOD	Successful interoperability test of Operational Test Fabric	Notification of successful Onboarding	Within 2 days of successful interoperability test

Table F1. Onboarding Process

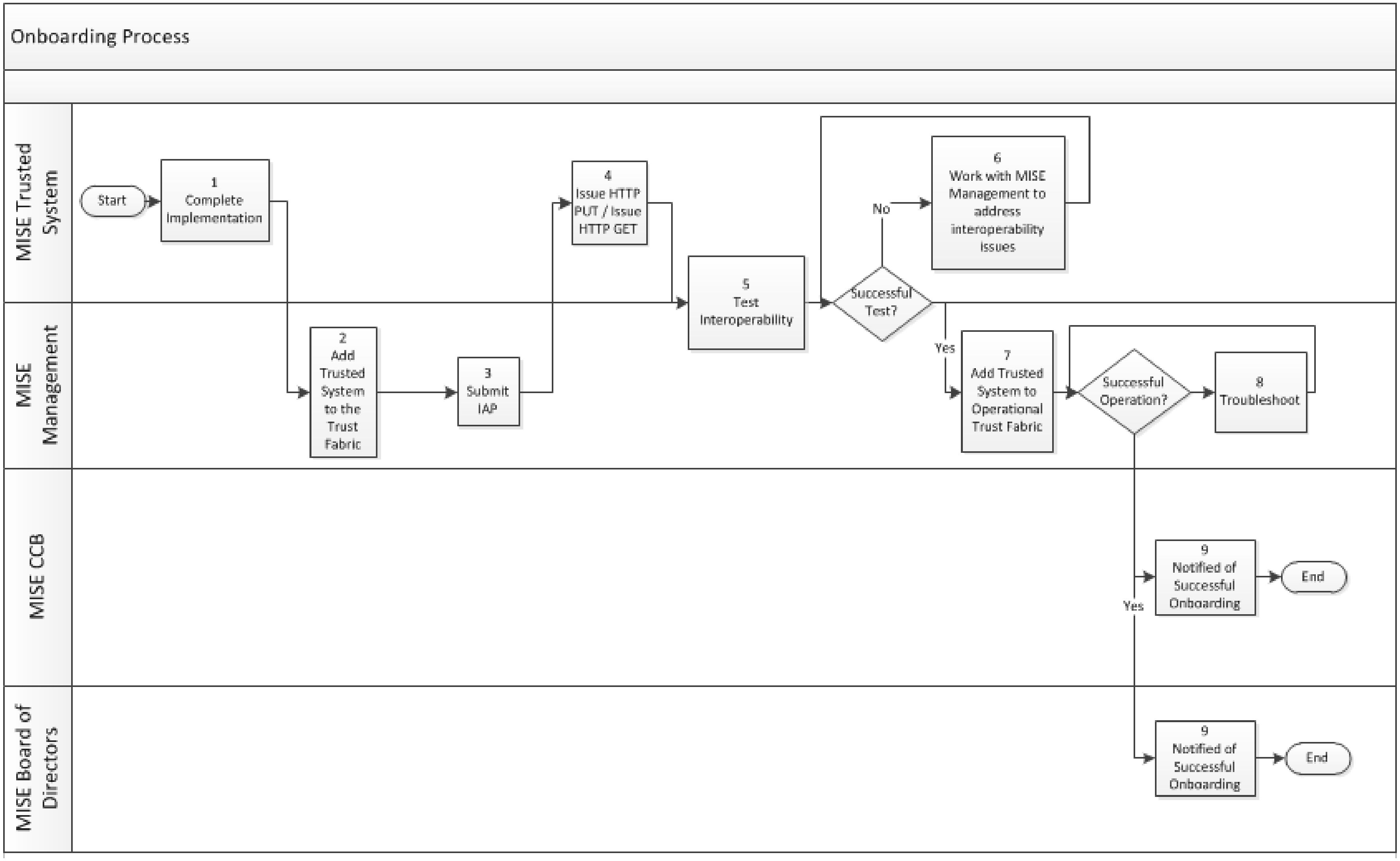

Onboarding Process
MISE Trusted System
MISE Management
MISE CCB
MISE Board of Directors
Start
1 Complete Implementation
2 Add Trusted System to the Trust Fabric
3 Submit IAP
4 Issue HTTP PUT / Issue HTTP GET
5 Test Interoperability
Successful Test?
No
Yes
6 Work with MISE Management to address interoperability issues
7 Add Trusted System to Operational Trust Fabric
Successful Operation?
8 Troubleshoot
Yes
9 Notified of Successful Onboarding
End
9 Notified of Successful Onboarding
End

Figure F1. Onboarding Process

MDA
ARCHITECTURE
MANAGEMENT
HUB

Version 2.0
Release 2
May 2013